HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

BACHELORARBEIT

Herr
Florian Kaiser

The Xaa-Proline *cis/trans* Isomerization in Globular Proteins: Extraction of Structural Features and the Development of a Support Vector Machine Based Prediction Tool

2012

Fakultät **MNI**

BACHELORARBEIT

The Xaa-Proline *cis/trans* Isomerization in Globular Proteins: Extraction of Structural Features and the Development of a Support Vector Machine Based Prediction Tool

Autor:
Florian Kaiser

Studiengang:
Biotechnologie/Bioinformatik (B. Sc.)

Seminargruppe:
BI09w3-B

Erstprüfer:
Prof. Dr. rer. nat. Dirk Labudde

Zweitprüfer:
Dipl.-Inf. Daniel Stockmann

Mittweida, August 2012

Bibliografische Angaben

Kaiser, Florian: The Xaa-Proline *cis/trans* Isomerization in Globular Proteins: Extraction of Structural Features and the Development of a Support Vector Machine Based Prediction Tool, 63 Seiten, 29 Abbildungen, 23 Tabellen, Software CD, Hochschule Mittweida (FH), Fakultät MNI

Bachelorarbeit, 2012

Satz: LaTeX
Druck: Auf ungebleichtem und damit umweltfreundlichem Papier.

Referat

In dieser Arbeit wurde eine Methode zur Vorhersage der Xaa-Prolin *cis*/*trans* Isomerie (Xaa ist eine beliebige Aminosäure) untersucht. Durch die Extraktion von zwölf strukturellen Merkmalen (echte Sekundärstruktur, innen/außen Klassifizierung von Prolin, Eigenschaften der Umgebung um Prolin und von Prolin selbst) wurde eine Support Vector Machine (SVM) als Methode zur Vorhersage umgesetzt. Die Java Software Xaa-PIPT wurde zur Extraktion der strukturellen Merkmale entwickelt. Basierend auf 4397 (2199 *cis* und 2198 *trans*) Prolinen aus nicht redundanten, globulären Proteinen wurde ein Klassifikator mit Hilfe des radialen Basisfunktion (RBF) Kernels trainiert. In zehnfacher Kreuzvalidierung erzielte dieser eine Genauigkeit von 70,0478 % und einen Matthews Korrelationskoeffizienten (MCC) von 0,4223. Außerdem wurde eine Sensitivität von 0,5433 und eine Spezifität von 0,8576 erreicht. Basierend auf diesem Klassifikator wurde eine leichtgewichtige und benutzerfreundliche Software in Java entwickelt (μXaa-PIPT), um die *cis*/*trans* Isomerie von Prolin vorherzusagen. Es wurde gezeigt, dass Korrelationen zwischen der räumlichen Umgebung von Prolin und der Isomerie bestehen. μXaa-PIPT kann für die Evaluierung von niedrig aufgelösten Proteinstrukturen und theoretischen Modellen verwendet werden, um deren Qualität durch die Vorhersage der Xaa-Prolin Isomerie zu verbessern.

In this work a new method for the prediction of the Xaa-proline (where Xaa is any amino acid) *cis*/*trans* isomerization was investigated. By extraction of twelve structural features (real secondary structure, inside/outside classification, properties of the environment around proline and proline itself) a support vector machine (SVM) based prediction approach was evolved. The Java software Xaa-PIPT for structural feature extraction was developed. Based on 4397 (2199 *cis* and 2198 *trans*) prolines extracted from non-redundant, globular proteins a classifier was trained using the radial basis function (RBF) kernel. In ten-fold cross-validation it achieved an accuracy of 70.0478 % and a Matthews correlation coefficient (MCC) of 0.4223, a sensitivity of 0.5433 and a specificity of 0.8576. Based on this classifier a lightweight and easy-to-use Java software tool, called μXaa-PIPT, for the prediction of the Xaa-proline *cis*/*trans* isomerization was developed. It was shown that there are correlations between the proline surrounding environment and the isomerization state. μXaa-PIPT can be used for the evaluation of low-resolution protein

structures and theoretical models to improve their quality by the prediction of the Xaa-proline isomerization.

I. Contents

II. List of Figures

III. List of Tables

IV. Nomenclature

5-HT_3	5-Hydroxytryptamine type 3 receptor, page 7
aa	amino acid, page 19
ACC	prediction accuracy, page 26
AUC	area under the ROC curve, page 27
Cdks	cyclin-dependent kinases, page 7
CFIS	chain folding initiation site, page 9
CSV	comma-seperated values, page 29
CTD	C-terminal domain, page 7
CUDA	Compute Unified Device Architecture, page 15
Dmp	5,5-dimethylproline, page 8
ECOC	error-correcting output code, page 14
EF-G	elongation factor G, page 49
ePros	Energy Profile Suite, page 22
FPR	false positive rate, page 27
GdnHCl	guanidine hydrochloride, page 9
GPU	graphics processing unit, page 26
GUI	graphical user interface, page 28
JRE	Java Runtime Environment, page 53
MCC	Matthews correlation coefficient, page 26
NMR	nuclear magnetic resonance, page 4
NTD	N-terminal domain, page 49
PDB	Protein Data Bank, page 1
PDBTM	Protein Data Bank of Transmembrane Proteins, page 16
PPIase(s)	peptidylprolyl isomerase(s), page 6
Pro	three-letter code of the amino acid proline, page 2
pSer	phosphorylated serine, page 6
PSI-BLAST	Position-Specific Iterative Basic Local Alignment Search Tool, page 47
PSSM	position specific scoring matrix, page 11
pThr	phosphorylated threonine, page 6
RBF	radial basis function, page 13
RNase A	ribonuclease A, page 8
ROC	receiver operating characteristics, page 27

SASA ..... solvent accessible surface area, page 18

SDK ..... software development kit, page 15

SVM ..... support vector machine, page 12

TNR ..... true negative rate, page 26

TPR ..... true positive rate, page 26

UniProtKB ..... UniProt Knowledgebase, page 49

Xaa ..... any amino acid, page 2

Xaa-PIPT ..... Xaa-Proline *cis/trans* Isomerization Prediction Tool, page 15

V. Acknowledgment

To all those, who just can't make money a priority. Don't stop.

Für Sara. Für meine Familie.

Ich danke besonders Alex für die tatkräftige Mit- und ausgezeichnete Zusammenarbeit an diesem interessanten und vielversprechenden Thema. Ich hoffe, dass wir noch einige Zeit gemeinsam forschen können und weiterhin so ein gutes Team bilden.

Mein Mentor, Herr Prof. Dr. rer. nat. Dirk Labudde hat den Grundstein zu dieser Arbeit gelegt. Er hat mich stets unterstützt und war hin und wieder der Quell für meine Motivation, wenn ich am Ende einer Sackgasse angelangte. Ich bewundere seinen Forschergeist und würde mich freuen, weiterhin mit ihm zusammenarbeiten zu dürfen, um meinen Beitrag zum Fortbestand und Erfolg der *bioinformatics group Mittweida*[1] leisten zu können.

Außerdem gilt Dank Herrn Dipl.-Inf. Daniel Stockmann, der dafür gesorgt hat, dass es nicht an nötiger Rechenleistung fehlte. Ohne ihn würden die Berechnungen vermutlich noch heute laufen.

[1] http://www.bioforscher.de

1 Introduction

1.1 Motivation

In summer 2011 the *cis*/*trans* isomerization of proline gained my interest. Prof. Dr. rer. nat. Dirk Labudde brought the project to my fellow student Alexander Eisold and me. He was also involved in former researches about this topic, namely:

- M. Schubert, D. Labudde, H. Oschkinat, and P. Schmieder.
  A Software Tool for the Prediction of Xaa-Pro Peptide Bond Conformations in Proteins Based on ^{13}C Chemical Shift Statistics.
  J. Biomol. NMR, 24(2):149–154, Oct 2002.
- D. Pahlke, C. Freund, D. Leitner, and D. Labudde.
  Statistically Significant Dependence of the Xaa-Pro Peptide Bond Conformation on Secondary Structure and Amino Acid Sequence.
  BMC Struct. Biol., 5:8, 2005.
- D. Pahlke, D. Leitner, U. Wiedemann, and D. Labudde.
  COPS – *cis*/*trans* Peptide Bond Conformation Prediction of Amino Acids on the Basis of Secondary Structure Information.
  Bioinformatics, 21(5):685–686, Mar 2005.

Nowadays, in times of high-throughput data processing in biology, even more information is available. In collaboration with Andreas Tzakos from the "Section of Organic Chemistry and Biochemistry" of the University of Ioannina[2] (Greece) it was our task to statistically analyze and collect data about the mechanism of proline isomerization. We developed software tools for the analysis of isomerization specific sequence patterns, the role of hydrogen bonds, the solvent accessibility and physicochemical properties of adjacent residues.

After these basic researches and analyses of proline isomerization, the idea was to develop an isomerization prediction tool for the evaluation of resolved protein structures. Furthermore there might be an application in the investigation of the very rare occurring *cis*/*trans* intermediate state of proline. Maybe, depending on the proline surrounding environment, assignments to this state can be made. In contrast to other already existing tools, which are using sequence or secondary structure information, this tool should be based on experimental determined structural features. Because there are currently over 83,000 entries in the Protein Data Bank (PDB), there would be enough data provided for the training of a machine learning algorithm. As a linear classifier the support vector

[2] http://www.uoi.gr/

machine would be very suitable to decide between the *cis* and the *trans* isomerization state.

1.2 The Xaa-Pro Peptide Bond Isomerization

1.2.1 The Dihedral Angle

The dihedral angle is defined as the angle between two planes, spanned by three bond vectors between four atoms. It describes the rotation of the bond as an interval reaching from -180 ° to 180 °. With the help of the dihedral angle it is possible to define conformations of molecules, such as the *cis* and the *trans* conformation. The *trans* conformation means the two considered molecule residues standing towards each other in relation to the reference plane, whereas the *cis* conformation says that they are on the same side of this plane.

The following explanation refers to the Ala-Gly-Pro example peptide (figure 1.1) and uses the same atom labels as seen in it. As the protein backbone consists of three repeating bonds (N-C$_\alpha$, C$_\alpha$-C' and C'-N) its conformation can be exactly described by the three dihedral angles of these bonds. The peptide bond linking two adjacent residues (C'-N) in a protein backbone can either occur in *cis* ($-30° \leq \omega \leq 30°$) or trans ($-180° \leq \omega \leq -150°$ and $150° \leq \omega \leq 180°$) conformation. The conformation determining dihedral angle is called the omega angle (ω). It is defined as the dihedral angle between the four atoms C$_\alpha$-C'-N-C$_\alpha$ (CA.2, CO.2, N.3 and CA.3) of the protein backbone. If ω does neither meet the assumptions for *cis*, nor for *trans*, there can be an intermediate *cis/trans* state assigned. Moreover the conformation of a protein backbone can be described by using the φ and ψ dihedral angles. The φ angle is used for the description of the N-C$_\alpha$ bond, including the atoms C'-N-C$_\alpha$-C' (CO.1, N.2, CA.2 and CO.2), whereas the ψ angle describes the rotation of the C$_\alpha$-C' bond with atoms N.2, CA.2, CO.2 and N.3 involved.

The calculation of the dihedral angle – for example ω – can be done as seen in equation 1.1. The advantage of this method is the two-argument $\mathrm{atan2}$ function (a variation of the arctangent function), which represents the angle with the correct sign. The dihedral angle is placed in the correct quadrant and is negative for clockwise angles ($y < 0$) and positive for counter-clockwise angles ($y > 0$). In informatics this is the most common and most efficient calculation method. This function is implemented in a large variety of programming languages.

$$v_1 = \begin{pmatrix} CO.2_x - CA.2_x \\ CO.2_y - CA.2_y \\ CO.2_z - CA.2_z \end{pmatrix} v_2 = \begin{pmatrix} N.3_x - CO.2_x \\ N.3_y - CO.2_y \\ N.3_z - CO.2_z \end{pmatrix} v_3 = \begin{pmatrix} CA.3_x - N.3_x \\ CA.3_y - N.3_y \\ CA.3_z - N.3_z \end{pmatrix}$$

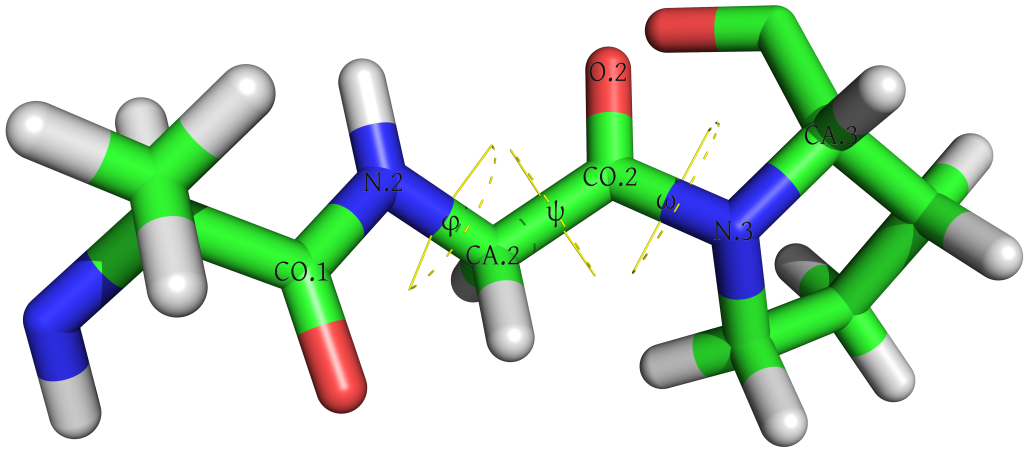

Figure 1.1: Example peptide Ala-Gly-Pro with φ, ψ and ω dihedral angles

$$\omega = \text{atan2}(x,y) = \text{atan2}(|v_2|v_1 \cdot [v_2 \times v_3], [v_1 \times v_2] \cdot [v_2 \times v_3]) \tag{1.1}$$

Where v_1 is the vector representing the bond between the atoms CA.2 and CO.2, v_2 representing the bond between CO.2 and N.3, v_3 representing the bond between N.3 and CA.3, respectively.

1.2.2 Xaa-Pro Isomerization

The peptide bond C'-N shows a partial double bond character, resulting form the nitrogen lone electron pair and the plane arrangement of the backbone atoms CA.2, CO.2, O.2, N.3, the hydrogen atom linked to N.3 (not present in proline) and CA.3 [Pahlke et al., 2005]. Due to this double bond character experimental and theoretical observations indicate a high energy barrier of around 20 $kcal \cdot mol^{-1}$ between the *cis* and the *trans* form of the C'-N bond. This barrier limits the rate of interconversion between these two forms. The *syn* transition state (ω = 90 °) has an energy that is significant higher than that of the *trans* state, which indicates that conversion from *cis* to *trans* state occurs very rarely [Lu et al., 2007]. Furthermore the interconversion rate is also solvent dependent and generally higher in polar solvents [Wedemeyer et al., 2002]. In 1951 LINUS PAULING proposed the following equation (1.2) to approximately determine the energy associated with the rotation around the C'-N peptide bond:

$$E = 30 \cdot \sin^2 \delta \qquad [kcal \cdot mol^{-1}] \tag{1.2}$$

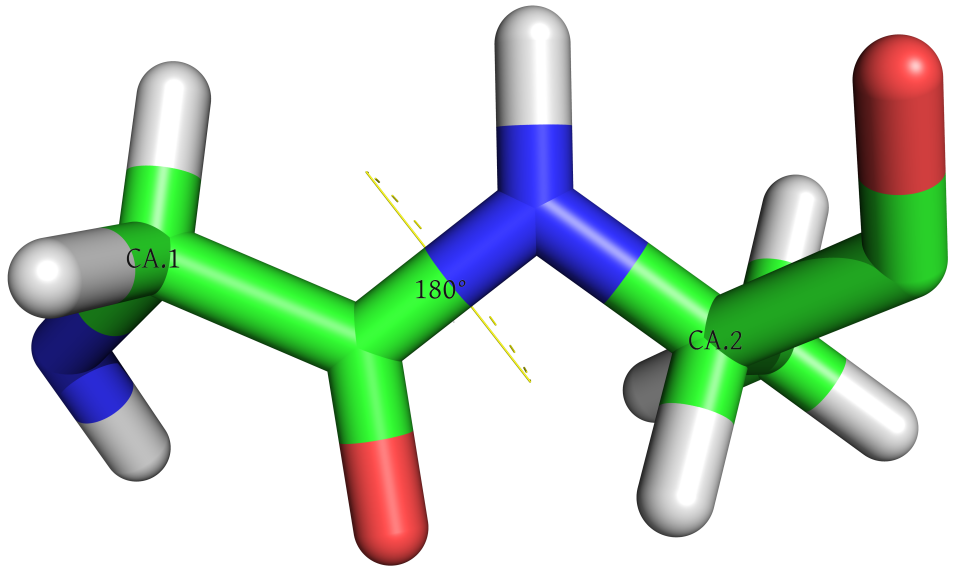

Figure 1.2: Example dipeptide Gly-Ala, *trans* conformation

Where δ is the dihedral angle measuring deviations from the absolute *trans* conformation ($\omega = 180°$). Recent analyses of high-resolution proteins confirmed the remarkable accuracy of this early approximation [Lu et al., 2007].

For all amino acid residues except proline the *trans* form (figure 1.2) shows a higher stability and is far more energetically favorable than the *cis* form. The instability of the *cis* form is caused by interactions between the two C_α atoms of the adjacent residues and the H atoms attached to them (figure 1.3).

Unlike other amino acids fragments the *cis* and *trans* forms of the Xaa-Pro (Xaa is any amino acid) fragment are almost isoenergetic. The energy barrier between the two forms in Xaa-Pro is reduced by unfavorable interactions between atoms present in both forms. This leads to a higher propensity for the *cis* conformation in Xaa-Pro fragments in relation to non-proline fragments. In fact, the *cis* conformation occurs with a frequency of 5-6 % in protein structures and a large majority of *cis* Xaa-Pro fragments occurs in bend and turn regions and typically solvent exposed on the surface of proteins [Pahlke et al., 2005].

Nuclear magnetic resonance (NMR) experiments showed that the *cis*/*trans*-ratio depends on the amino acid sequence adjacent to the proline. Correlations between the bulkiness of the sidechain of a preceding residue and the isomerization rate were figured out: the higher the bulkiness of the residue, the lower the isomerization rate is. Preceding aromatic residues for example, can cause approximately a ten-fold reduction of the isomerization rate in comparison to alanine. Further studies also showed a connection between the *cis*/*trans*-ratio and the nature of the proline succeeding residue.

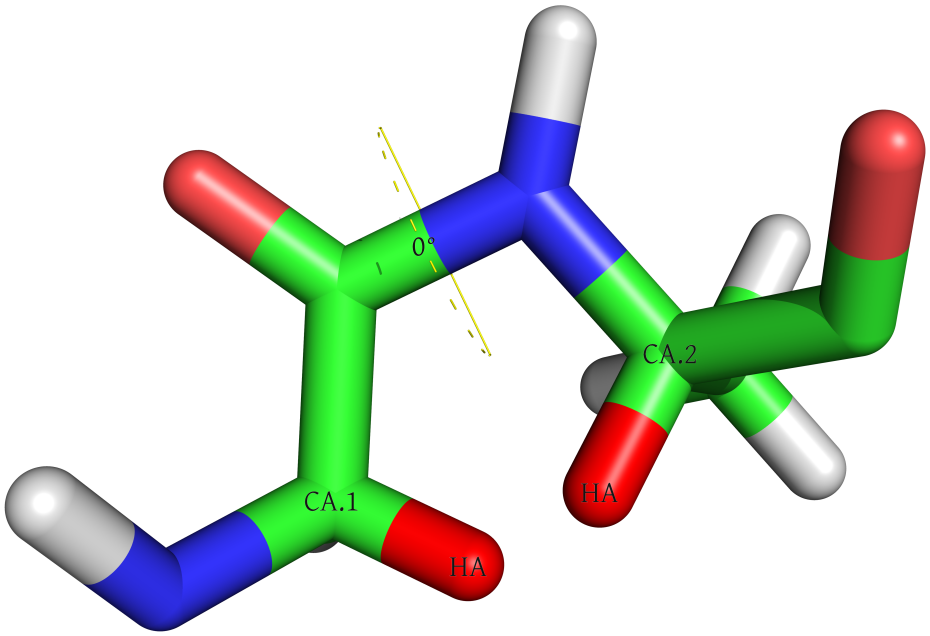

Figure 1.3: Example dipeptide Gly-Ala, *cis* conformation – unfavorable interactions between C_α atoms and hydrogen atoms HA (red highlighted) in *cis* conformation

Positively charged sidechains seem to destabilize the *cis* conformation relative to *trans*. In contrast to this, aspartate, asparagin and glycine stabilize the *cis* form [Pahlke et al., 2005].

1.2.3 Biological Relevance of the Xaa-Pro Isomerization

The isomerization of Xaa-Pro was figured out to play a widely spread role in biochemical processes in the cell. Currently only a few of the possible functions were experimental or theoretical confirmed.

Studies showed that proline residues – existing in two completely distinct conformations – provide a switch in the protein backbone, controlled by prolyl *cis*/*trans* isomerization. Structural differences between the *cis* and the *trans* form act as a molecular switch, toggling for example two functional states of the protein. This can determine for example different sets of intermolecular binding partners. The high energy barrier between these two forms imparts a very slow timescale of uncatalyzed interconversion, usually many minutes. Rate measurements by NMR and by proteolytic or protease free assays confirmed slow *cis*-to-*trans* conversion rates around 0.002 s^{-1} at 25 °C for Xaa-Pro [Lu et al., 2007]. This is outstanding and isolates the isomerization of Xaa-Pro from the fast timescales of biochemical reactions in the cell [Lu et al., 2007]. Without a catalyst the interconversion between the *cis* and *trans* form happens very slowly. However, the isomerization of the peptide bond can be generally catalyzed by disrupting its partial double bond character. This happens for example if a strong acid is present, which

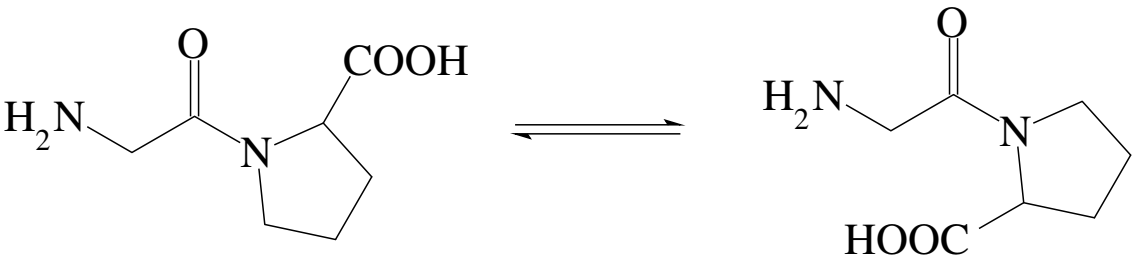

Figure 1.4: Example dipeptide Gly-Pro, isomerization process

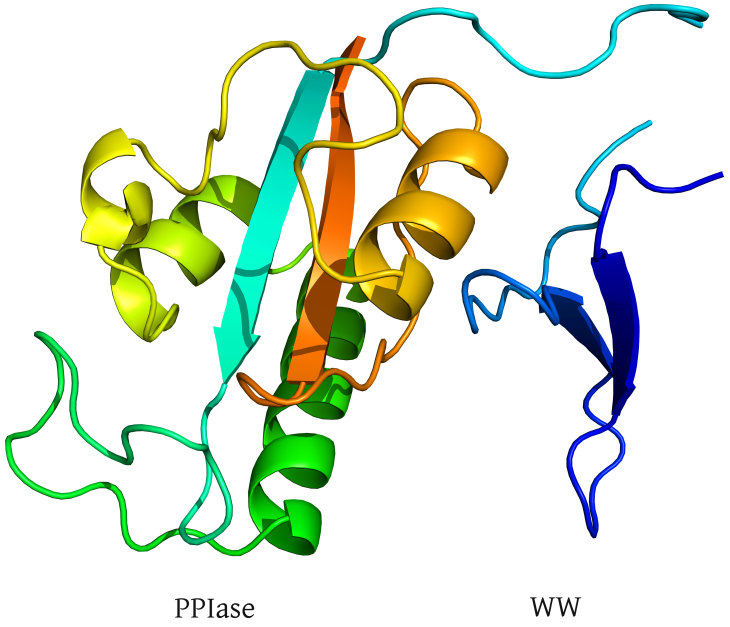

Figure 1.5: Human Pin1 structure with two distinct domains – X-ray diffraction, 1.35 Å resolution, PDB-ID 1PIN

then increases the interconversion rate by protonation of the carbonyl oxygen atom of the amino acid [Wedemeyer et al., 2002]. Beside this, there are also some enzymes – called peptidylprolyl isomerases (PPIases) – which can act as a catalyst of the *cis*/*trans* isomerization. PPIases can catalyze the interconversion between the *cis* and the *trans* form in both directions (see figure 1.4).

The human Pin1 PPIase, an phosphorylation-dependent PPIase, binds to and isomerizes specific phosphorlyated Ser/Thr-Pro motifs. Pin1 has two separate subunits for a binding (WW) and a catalytic function (PPIase) as seen in figure 1.5.

The enzyme uses substrate phosphorylation as an additional level of regulation: only when the substrate is phosphorylated the *cis*/*trans* isomerization is catalyzed by Pin1. The isomerization of pSer/pThr-Pro motifs is especially significant because proline-directed kinases and phosphatases are conformation specific, acting only in *trans* conformation. Furthermore phosphorylation slows down the natural isomerization rate of the Ser/Thr-Pro peptide bond and makes it resistant against other PPIases [Lu et al.,

2007]. This lead to the hypothesis of a new cell signaling mechanism, whereby Pin1 catalytically regulates the conformation of its substrate after phosphorylation to control protein function. This can have impact on many key proteins in diverse cellular processes, such as cell growth regulation or stress responses. Pin1 emerged as a molecular timer, controlling the amplitude and the duration of a cellular process under a given condition. As this is a very tightly regulated process, deregulation plays a critical pathological role in aging or diseases like cancer, Alzheimer's disease and asthma [Lu et al., 2007]. Also the progression through different phases of the cell cycle is regulated by activation and inactivation of several proline-directed, cyclin-dependent kinases (Cdks). It is not clear how the proteins, phosphorylated by those kinases, are coordinated for highly choreographed cell cycle events. Recent results suggest that posttranslational phosphorylation through Pin1 might be an important mechanism for the coordination of mitosis. In fact Pin1 affects mitotic progression in yeast and cancer cells [Lu et al., 2007]. Further studies have demonstrated a critical role for prolyl *cis*/*trans* isomerization in determining the timing and the duration of diverse signal pathways, involved in cell proliferation and transformation. A well documented example is the influence of phosphorylation-specific prolyl isomerization in amplifying the Neu-Raf-Ras-MAP kinase pathway (growth factor induced cell cycle progression) at multiple levels [Lu et al., 2007]. Beyond this, phosphorylation-specific prolyl isomerization plays a regulatory role in cell signaling, ion channel gating [Lummis et al., 2005] and phage infection. Pin1 catalyzed phosphorylation-dependent isomerization has also been shown to regulate gene-expression. The function of many transcription factors is regulated directly via many different mechanisms. Pin1 can act as regulatory mechanism for RNA polymerase II. It binds to pSer-Pro motifs in the C-terminal domain (CTD) of the large subunit of RNA polymerase II and stimulates its dephosphorylation by Fcp1 phosphatase. It has been reported that Pin1 increases phosphorylation of CTD or inhibits dephosphorylation in mammalian cells and might be involved in transcriptional suppression during mitosis [Lu et al., 2007]. The ubiquitous presence of PPIases and their widely spread roles underscore the biological importance of Xaa-Pro *cis*/*trans* isomerization. It can be seen as a fundamental molecular switch, controlling the timing of many biological key processes. The *cis* and *trans* isomers provide stable local and strong differing structures and thereby a mechanism for proteins to select different pools of binding partners even in otherwise unstructured regions of the protein [Lu et al., 2007].

Concerning the previously mentioned role of Xaa-Pro isomerization in ion channel gating, LUMMIS ET AL. published extensive researches about the 5-hydroxytryptamine type 3 receptor (5-HT_3), a neurotransmitter-gated ion channel. They figured out that the highly conserved Pro8, which is located at the apex of the loop between the second and third transmembrane helices of 5-HT_3, is essential for ion gating. The replacement of Pro8 with Gly, Ala, Cys, Val, Lys or Asn resulted in correct folded receptors with antagonist binding functionality. Nevertheless all were non-functional, which indicates the essential role of Pro8 in ion channel gating [Lummis et al., 2005]. Figure 1.6 shows the Pro8 in a similar protein structure (nicotinic acetylcholine receptor) to illustrate the issue.

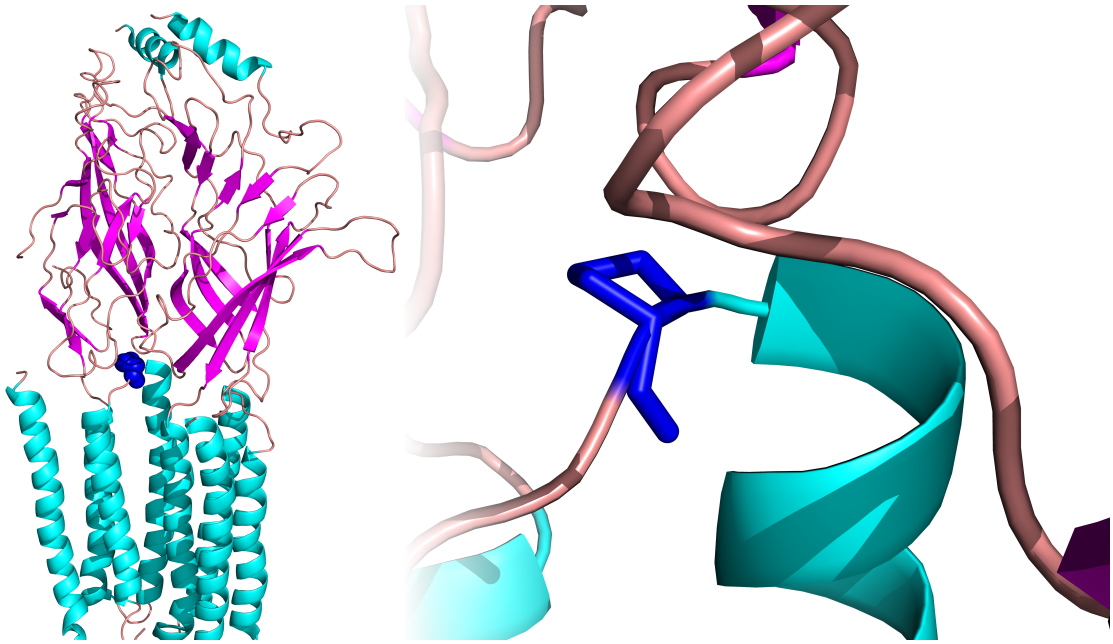

Figure 1.6: Refined structure of the nicotinic acetylcholine receptor, 4 Å resolution, Pro8 blue highlighted as sphere model in overview (left) and as stick model in detail (right), PDB-ID 2BG9

The chains D and E in overview are shown left and Pro8, located at the apex of the loop between the second and third membrane spanning helices, in detail (right).

By substitution of prolines at other positions by artificial compounds like α-hydroxyvaline, which is not a hydrogen bond donor, a functional receptor was produced. However, by substituting Pro8 with other cyclic compounds or compounds without a hydrogen donor function, mixed results were produced. The key was found in compounds with different preferences for the *cis* conformation. If strongly *trans* conformation favored compounds were inserted instead of Pro8, this lead to non-functional receptors. Whereas some *cis* favored compounds even increased the activity of the receptor significantly. The most extreme result yielded 5,5-dimethylproline (Dmp) which strongly prefers the *cis* conformation. With Dmp the half maximal effective concentration (EC_{50}) was decreased up to 60-fold [Lummis et al., 2005]. The researches of LUMMIS ET AL. emphasized the important role of the Xaa-Pro *cis*/*trans* isomerization in neuronal biochemical signal transduction processes.

1.2.4 Influence of the Xaa-Pro Isomerization on Protein Structure

Studies showed that the isomerization of the Xaa-Pro peptide bond is one of the rate determining steps in the folding process of proteins. WEDEMEYER ET AL. analyzed the correlation between conformational changes of the folding process and the *cis*/*trans* isomerization of Xaa-Pro in disulfide-intact bovine pancreatic ribonuclease (RNase A) *in vitro*. As RNase A has four prolines (Pro93, Pro114, Pro42 and Pro117) it is a good

model protein for studying the coherences. In native state the Pro93 and the Pro114 are in *cis*, the remaining in *trans* conformation. The goal of this study was to determine how each of the 2^4 possible *cis*/*trans* species in the unfolded protein influences the correct folding process. And afterwards compare these results with the structural disruption caused by non-native isomers of RNase A. By unfolding the protein with guanidine hydrochloride (GdnHCl) and all possible single proline-to-alanine mutations the essential prolines were figured out [Wedemeyer et al., 2002]. Pro42 was identified to be the only non-essential proline in bovine RNase A. This was not expected, as Pro42 occurs in a highly conserved and central β-strand region. After determining the essential prolines, correlations between these and proline *cis*/*trans* isomerization rate were examined [Wedemeyer et al., 2002]. The research showed that a change in a single dihedral angle should be capable to completely disrupt the natural folding process. A possible explanation is that non-native proline isomers disrupt conformational folding by altering critical hydrogen bonds between amino acid residues, which are essential for correct formation of secondary structure elements. This leads to a often up to 1000-fold slower folding process or the complete disruption of folding. For example Pro93 in bovine RNase A (native in *cis* conformation) naturally forms a β-turn, may acting as chain folding initiation site (CFIS), which catalyzes the formation of the two-stranded β-hairpin (residues 79-104). The non-native Pro93 isomer disrupts the native turn, leading to a larger loop, which needs to be closed to form the correct hairpin and therefore lowering the native turns formation rate [Wedemeyer et al., 2002].

Proline as itself has some general influences on the conformational structure of the protein. At first hydration is favored at proline residues. Secondly proline occurs frequently at corner positions in β-turns and at the end of strands and helices. Furthermore it has helix-breaking properties. Due to this fact structural reorganization during Xaa-Pro *cis*/*trans* isomerization, either in proximity of proline or at distant positions, is critical. Likewise it is critical for the relationship between the folding state and protein activity. As already demonstrated, Xaa-Pro *cis*/*trans* isomerization has a significant influence on this. The rate of isomer-specific protein-ligand interactions is often high (>10^3-fold) [Reimer and Fischer, 2002].

REIMER ET AL. have done some researches about how the consequences of different native Xaa-Pro conformations in homologous proteins are. They analyzed the consequences of isomerization for the structure around the specific proline and for positional preferences of particular amino acids. This was done by computational comparing the distances between C_α atoms of the protein backbone. Additional a statistical analysis of non-homologous proteins had been done to eventually predict the maximal distances, at which changes will occur as a result of prolyl isomerization [Reimer and Fischer, 2002]. For all 20 amino acids Xaa at position i a dataset, containing the following distance information, was created: C_α $(i-3)$ to C_α (i), C_α $(i-2)$ to C_α (i), C_α $(i-1)$ to C_α $(i+2)$ and C_α $(i-1)$ to C_α $(i+1)$ [Reimer and Fischer, 2002]. The results showed that for 19 of 20 amino acids the statistical distribution pattern of C_α to C_α distances were

identical in N- and C-terminal direction. Interestingly, if Xaa is proline, the distances in both directions showed significant differences, clustering in two groups, depending on if Xaa-Pro was in *cis* or *trans* conformation. For example the differences in C_α $(i-1)$ to C_α $(i+2)$ distances are very obvious. For *trans* proline at position i the distance is approximately about 6 Å, whereas it is about 5 Å for *cis* proline. This suggests that there is a high probability for an expansion of the protein backbone if *cis* isomerizes to *trans* conformation. In contrast no difference in distances depending on Xaa-Pro *cis*/*trans* isomerization could be found for C_α $(i-3)$ to C_α (i) and C_α $(i-2)$ to C_α (i). This leads to the conjecture that almost no conformational effects of Xaa-Pro *cis*/*trans* isomerization happen in N-terminal direction of the protein backbone. In fact prolyl isomerization affects remote peptide segments asymmetrically. However, for non-homologous proteins it is not possible to determine isomer-specific effects exactly, because there is no clear separation from effects caused by differences in the amino acid sequence [Reimer and Fischer, 2002].

Out of 1699 proteins with at least 95 % sequence identity it was possible to find 64 proteins with a *cis*/*trans* distinction in a single Xaa-Pro peptide bond. Positional preferences of amino acids surrounding the isomeric Xaa-Pro bond were identified. The values were normalized by dividing the occurrence at a particular position by the statistically percentage of the amino acid in the Protein Data Bank (PDB). For position $i-3$ phenylalanine, proline and tyrosine occurred at least two-fold the expected value. At position $i-2$ a preference of tryptophan was measured, at $i-1$ asparagine, glycine and serine occurred more often. In C-terminal direction cysteine, phenylalanine and tryptophan showed a higher occurrence at position $i+1$, methionine at position $i+2$ and proline at position $i+3$ [Reimer and Fischer, 2002].

1.3 Dependence of Structural Information

The mechanism of Xaa-Pro isomerization is visible in different levels of information of a protein. There exist basically three levels of information: the one-dimensional or sequence based information, the two-dimensional secondary structure information and the three-dimensional structural information. Every ascending level of information gets more complex.

Correlations between the *cis*/*trans*-ratio of proline peptides and the primary amino acid sequence are well known and proven [Grathwohl and Wüthrich, 1981]. The occurrence of specific sequence patterns and residues, adjacent to proline, affect the isomerization. Also the influence of secondary structure elements on the isomerization has been studied and confirmed [Pahlke et al., 2005]. These information were already used for prediction approaches. PAHLKE ET AL. developed a software for the prediction of the isomerization state of proline in 2005. They used sequence information, such as the occurrence of other amino acids preceding and succeeding to proline and secondary

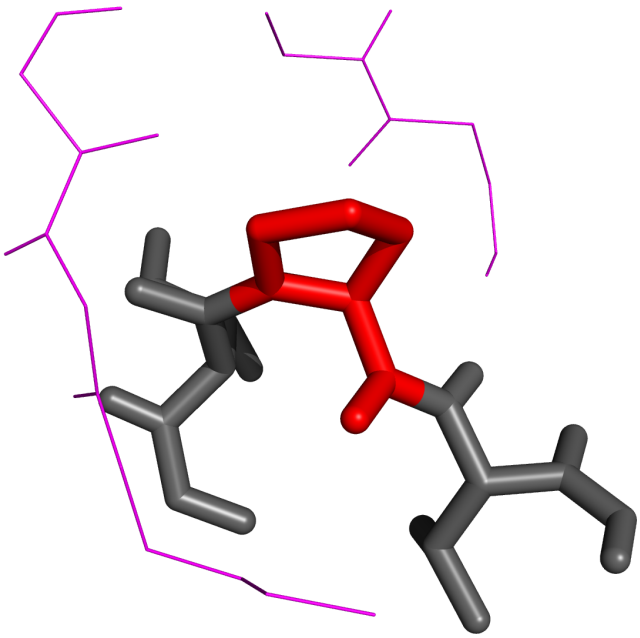

Figure 1.7: *cis*-proline 419 in *Escherichia coli* HSP70 chaperone – NMR, PDB-ID 2KHO

structure information derived from hydrogen bonding patterns by the DSSP[3] algorithm. The results showed a significant impact of secondary structure on Xaa-Pro isomerization and a slightly influence of residues adjacent to proline. SONG ET AL. developed a web server (CISPEPpred[4]) for the prediction of proline peptide bonds in 2006 [Song et al., 2006]. They used multi sequence alignment information in form of position specific scoring matrices (PSSMs) and secondary structure information to predict the isomerization of Xaa-Pro with the single amino acid sequence. In 2008 EXARCHOS ET AL. published a method for the prediction of the *cis*/*trans* isomerization of peptides (including non-proline peptides) based on evolutionary profiles, secondary structure information, solvent accessibility and physicochemical properties of proline surrounding residues [Exarchos et al., 2009].

The real structural information, present in high-resolution protein structures, was not yet used for feature extraction. Figure 1.7 shows the proline at position 419 in the bacterial HSP70 chaperone. This proline was determined to be in *cis* conformation. The gray colored amino acids are representing the backbone, whereas the magenta colored residue fragments are less than 5 Å away from the C_α atom of proline. In this work the main approach was the meaningful extraction of such structural information in the environment of the proline of interest. There are compulsory coherences between the isomerization state of proline and the surrounding environment. Not at least because of the 1 Å shorter distance of proline in *cis* conformation, which leads to a higher density of the environment for the *cis* case. In figure 1.8 the *trans* proline at position 142 clearly indicates the less dense information within the 5 Å environment

[3] http://swift.cmbi.ru.nl/gv/dssp/
[4] http://sunflower.kuicr.kyoto-u.ac.jp/ sjn/cispep/

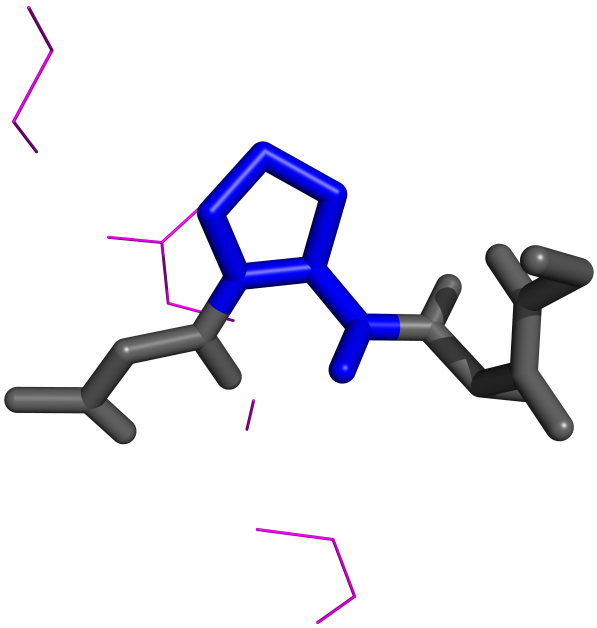

Figure 1.8: *trans*-proline 142 in *Escherichia coli* HSP70 chaperone – NMR, PDB-ID 2KHO

This environment can be analyzed in respect to many different properties. More specifically the bulkiness of proline preceding residues was figured out to have an impact on proline *cis*/*trans* isomerization [Pahlke et al., 2005]. This possibly can be adapted to the proline environment.

Beside the proline surrounding residues the real secondary structure can be obtained from the protein structure. As already mentioned this feature has a large impact on the isomerization of proline residues. The quality of the real secondary structure exceeds the accuracy of predicted secondary structure elements and therefore provides an important feature for the prediction.

1.4 Support Vector Machine Classification

There are several machine learning techniques for various purposes. The support vector machine (SVM) can learn from empirical data to be then applied on unknown data for classification and regression problems. In this work the SVM was used as a classifier for the two-class problem of *cis* and *trans* isomerization of Xaa-Pro. Because of the strictly theoretical background of SVMs they were not widely appreciated at first. The first publications were released in the 1960s by VAPNIK, CHERVONENKIS and coworkers, but remained mainly unnoticed until the early 1990s. Beside the theoretically applications, there were no practical ones recognized. By now SVMs achieved better results than neural networks and other statistical learning algorithms on the most popular benchmark problems [Kecman, 2005].

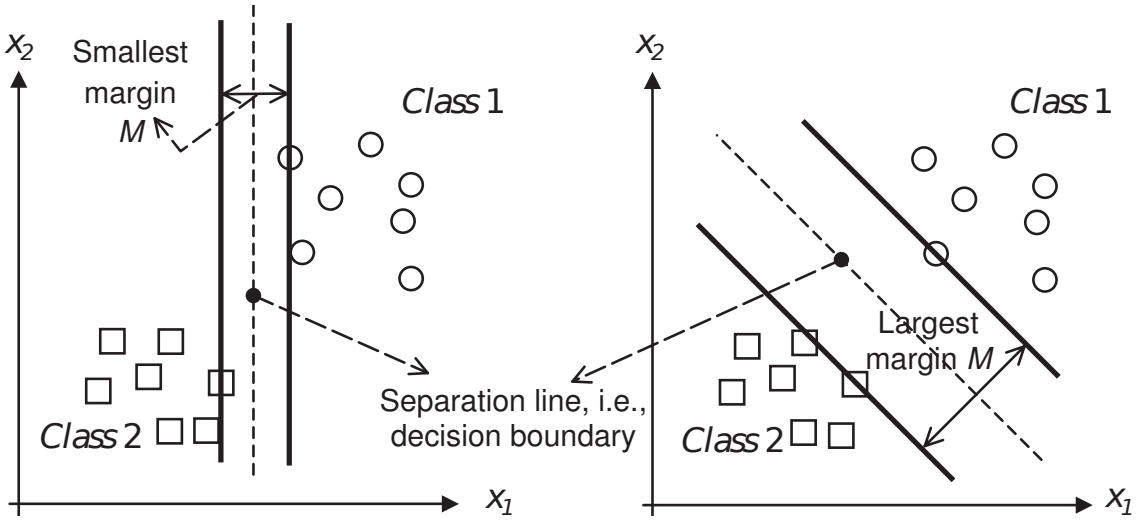

Figure 1.9: Two-out-of-many separating lines: a good one with a large margin (right) and a less acceptable separating line with a small margin (left) [Kecman, 2005]

The following learning problem is defined for SVMs:

Theorem 1.1 *There is some unknown, non-linear dependency* $y = f(x)$ *between a high dimensional input vector* x *and a scalar output* y *[Kecman, 2005].*

As there is no information about the underlying probability functions, SVMs must perform a so called distribution-free learning. There is only information available about the training data set. Furthermore the SVM belongs to the supervised learning techniques. For simple classification tasks the SVM uses a linear separating hyperplane to create a classifier with a maximal margin between the two classes (see figure 1.9). The penalty parameter C determines the margin size for misclassified data points. A large C leads to less misclassified data, a smaller margin and vice versa. If the data is not linear separable in its original input space, the machine transforms the input space into a higher-dimensional feature space [Kecman, 2005]. This non-linear transformation can be achieved by using various non-linear mappings. Accordingly the SVM implements so called kernel functions to achieve these transformations. If the classes are transformed into the higher feature space, they can become linearly separable. There are several kernel functions existent, but mainly three are common basic: linear kernels, polynomial kernels and radial basis function (RBF) kernels.

$H(x,x')$ is defined as the kernel function of pairs of training vectors x_i and x_j. If the classes are linearly separable in input space, there is no need for mapping them into a higher-dimensional feature space. Thus linear kernels are used in this situation. Table 1.1 shows the three basic kernel functions [Abe, 2005b].

Table 1.1: SVM basic kernel functions [Abe, 2005b]

kernel	**function**	**parameters**
linear	$H(x,x') = x^T x'$	-
polynomial	$H(x,x') = (x^T x' + 1)^d$	d
RBF	$H(x,x') = \exp(-\gamma \|x - x'\|)^2$	γ

SVMs are formulated for two-class classification problems. Therefore an implementation as a multi-class capable classifier is not trivial. To give a short overview of the potential of SVMs, the four types, which can handle multi-class problems, are listed below.

1. one-against-all SVMs
2. pairwise SVMs
3. error-correcting output code (ECOC) SVMs
4. all-at-once SVMs

One-against-all SVMs divide the n-class problem into n two-class problems where the ith class of the ith two-class problem is separated from the other classes. Pairwise SVMs are converting the n classes into $\frac{n(n-1)}{2}$ classes, therefore all pairs of classes are covered. The ECOC SVMs can be used to resolve non-classifiable regions of multi-class problems. With all-at-once SVMs all decision functions for classification are determined simultaneously [Abe, 2005a].

2 Methods

The following sections describe the methods applied in this work. The computing and programming related steps are described for Unix-like systems. The software was programmed in Java, because of cross-platform availability. The used software development kit (SDK) was Eclipse 4.2 Juno. For simplicity the BioJava 3.0.4 open-source framework was used. This framework provides countless of powerful analysis and statistical routines for the processing of biological data. It can read common file formats and implements standard alignment tools, sequence and 3D structure manipulation packages [Holland et al., 2008]. The SVM implementation libSVM 3.12 by CHANG and LIN was used for the realization of the classifier [Chang and Lin, 2011]. The package is available for a wide variety of programming languages. It offers a Java version and even a reimplementation for CUDA acceleration.

As a main work package of this project the software "Xaa-Proline *cis/trans* Isomerization Prediction Tool" (Xaa-PIPT) , mentioned and described in subsection 2.10.1, was developed and used for most of the described methods in the following sections. It emerged to a tool for feature extraction and building a SVM based classifier for Xaa-Pro isomerization. Based on the with Xaa-PIPT built classifier, μXaa-PIPT (described in subsection 2.10.2) was developed as an easy-to-use prediction tool for the isomerization state of prolines in PDB structures based on structural features.

2.1 Creating a Representative Dataset

The first step was the definition of a representative dataset of globular proteins, which can be found on the attached CD (content see appendix A). For the collection of the protein structures the PISCES[5] web server was used [Wang and Dunbrack, 2003]. This protein sequence culling server can obtain protein chains from the PDB regarding predefined parameters. The results are received as a list of PDB- and chain IDs. Beside this, the corresponding sequences in FASTA format can be obtained simultaneously. Table 2.1 shows the applied criteria for the PISCES server.

The PISCES server culled 7779 protein chains meeting the specified criteria. The dataset provides non-redundant protein chains to guarantee a large diversity of training vectors for the SVM. Furthermore only high-resolution proteins are observed. This is due to the exact representation of the Xaa-Pro peptide bond isomerization. Low resolution proteins do not offer such exact information. A R-factor less than 0.3 ensures that the protein chain models fit their X-ray diffraction data quite well, which is also an important fact for the quality of the training data.

[5] http://dunbrack.fccc.edu/PISCES.php/

Table 2.1: PISCES criteria for the creation of the dataset

criteria	**value**
sequence percentage identity	$\leq$ 25 %
sequence chain length	40 - 10000
resolution	0.0 - 2.4 Å
R-factor value	0.3
non-X-ray entries	include
C_α-only entries	exclude
cull PDB by	chain

Table 2.2: Data preprocessing, overview

dataset	**number of entries**	**relative**
PISCES	7779	100 %
redundant to PDBTM	58	0.75 %
final dataset	7721	99.25 %

2.2 Dataset Preprocessing

The dataset obtained from the PISCES web server did not only contain globular proteins, but also transmembrane proteins. It was necessary to clean the dataset and remove all transmembrane proteins. A list of non-redundant transmembrane proteins was taken from the Protein Data Bank of Transmembrane Proteins (PDBTM). This data bank is redundant with the PDB but contains only transmembrane proteins. There were 58 transmembrane chains found in the initial PISCES dataset and then removed. Table 2.2 shows detailed information about the final dataset.

Consequently 7721 protein chains were suitable for the search for prolines and structural feature extraction.

2.3 Extraction and Abstraction of Structural Features

As mentioned in section 1.3 the extraction of useful structural information around the Xaa-Pro fragment of interest was the main task in this work. Furthermore the abstraction of the gained information had to be done to fit as SVM input. The SVM only accepts discrete numeric values as input for features, where multi-categorical features can be represented as vectors in binary format.

The following properties of proline and the environment around proline were determined:

Table 2.3: Abstraction of secondary structure elements

secondary structure element	abstracted value for SVM input
α-helix	-1
β-strand	0
coil	1

- author assigned secondary structure of proline (i) and the amino acids at $i-2$, $i-1$, $i+1$, $i+2$
- inside/outside classification of proline
- environment hydrophobicity
- environment polarity
- environment mutability
- environment bulkiness
- environment flexibility
- proline energy

The calculation methods of each feature are described in the following subsections. If one of the features could not be calculated (e. g. because of missing atoms), the current proline was excluded from further calculations due to lack of information. Only completely calculated prolines were used for the training of the SVM.

2.3.1 Real Secondary Structure

The secondary structure assignment of a PDB structure is essential and thus present in all resolved structures. The assignment is done by the author(s) of the PDB structure and matches with the experimental observations. For the extraction of the secondary structure information the BioJava method

```
org.biojava.bio.structure.AminoAcid.getSecStruc()
```

was used.

Beside the secondary structure of proline itself (at position i), the secondary structures of the amino acids at the positions $i-2$, $i-1$, $i+1$ and $i+2$, relative to proline, were determined. The secondary structure was then classified into helix, strand and coil. Table 2.3 shows the abstraction of secondary structure elements used as SVM input.

2.3.2 Inside/Outside Classification

The calculation of the solvent accessible surface area (SASA) of proteins can be done relatively straightforward. It is usually measured in Å^2 and was firstly described by LEE and RICHARDS in 1971 [Lee and Richards, 1971]. The Shrake-Rupley algorithm, introduced in 1973, can be used for the calculation of the SASA. Firstly the van der Waals surface of the atoms is given by the atomic radii. Secondly a probe sphere of a defined radius (representing the solvent, e. g. 1.4 Å for water) is rolled along the surface. The track of the center of the probe then builds the SASA [Shrake and Rupley, 1973].

However, there are some deficiencies of this algorithm. One of the major disadvantage is, that it cannot determine the SASA of transmembrane proteins correctly. This is because of the membrane spanning regions, which are exposed to the hydrophobic membrane and therefore not solvent accessible. But nevertheless they are determined as solvent accessible by the Skrake-Rupley algorithm. A possible alternative is the inside/outside classification of amino acids. Theorem 2.1 shows how the inside/outside state of an amino acid is determined [Heinke and Labudde, 2012].

Theorem 2.1 *An amino acid n is classified as **inside** if the position of the C_β atom of n and the C_α atom of n and the geometric centroid c of all C_α atoms within a sphere of 10 Å around n fulfills:*

$$|C_\alpha - c| \leq 5 \text{ Å} \cup (C_\beta - C_\alpha)(c - C_\alpha) \leq 0$$

*Otherwise n is classified as **outside**.*

For the amino acid glycine, which does not have a C_β atom, it is approximately determined. One possibility to do this is the usage of the static BioJava method

```
org.biojava.bio.structure.Calc.createVirtualCBAtom(AminoAcid amino).
```

This inside/outside classification approach was applied on every proline. For the necessary abstraction *inside* was defined as `-1` and *outside* as `1`.

2.3.3 Environment Around Proline

The environment around proline can be quite well described within a sphere of a defined radius. So all amino acids surrounding the proline of interest can be observed independently of being covalent linked with proline. The amino acids lying in this sphere can then be investigated concerning different physicochemical and other properties. The

best radius was evaluated to be 5 Å (see section 3.2), so this radius was used in all described methods.

However, there need to be an abstraction of the properties done. An easy and effective method is the use of amino acid property scales. These scales allocate each amino acid type a numeric value, depending on the characteristics of the amino acid. Over the years lots of scales had been described by researchers. ProtScale, a summary of such scales, can be found on the ExPASy server[6]. ProtScale is not just a database for scales of amino acids, it also can calculate amino acid properties in defined window sizes of given protein sequences.

To avoid distortion of the results for spheres with very few amino acids within it (for example for prolines at the surface of proteins), the sphere density ρ as normalization parameter is applied. The density is defined as seen in equation 2.1.

$$\rho = \frac{n_{aa}}{\frac{4}{3}\pi r^3} \qquad [aa \cdot \mathring{A}^{-3}] \tag{2.1}$$

Where n_{aa} is the quantity of all amino acids within the sphere and r is the sphere radius. Figure 2.1 shows the issue in detail. The HSP70 chaperone Pro134 is surface-exposed and thus the 5 Å environment around it does not contain many amino acids (shown in magenta). Likewise this is often the case for surface-exposed amino acids, whereas Pro172 – which is a buried residue – has much more amino acids around it. This underscores the importance of the normalization parameter ρ to avoid environments containing only a few amino acids, to have the same weight as high dense environments.

Hydrophobicity The scale defined by KYTE and DOOLITTLE in 1982 was used for the abstraction of the environment hydrophobicity [Kyte and Doolittle, 1982]. Table 2.4 contains the values assigned to each amino acid. A negative value represents hydrophilic amino acids and a positive value hydrophobic amino acids, respectively. Equation 2.2 shows how the hydrophobicity of the environment (H_{env}) around proline containing n amino acids is abstracted, where $H(aa_i)$ is the numeric value for the amino acid aa_i.

Table 2.4: Hydrophobicity scale by KYTE and DOOLITTLE (1982)

aa	**Ala**	**Arg**	**Asn**	**Asp**	**Cys**	**Gln**	**Glu**	**Gly**	**His**	**Ile**
H(aa)	1.8	-4.5	-3.5	-3.5	2.5	-3.5	-3.5	-0.4	-3.2	4.5
aa	**Leu**	**Lys**	**Met**	**Phe**	**Pro**	**Ser**	**Thr**	**Trp**	**Tyr**	**Val**
H(aa)	3.8	-3.9	1.9	2.8	-1.6	-0.8	-0.7	-0.9	-1.3	4.2

[6] http://web.expasy.org/protscale/

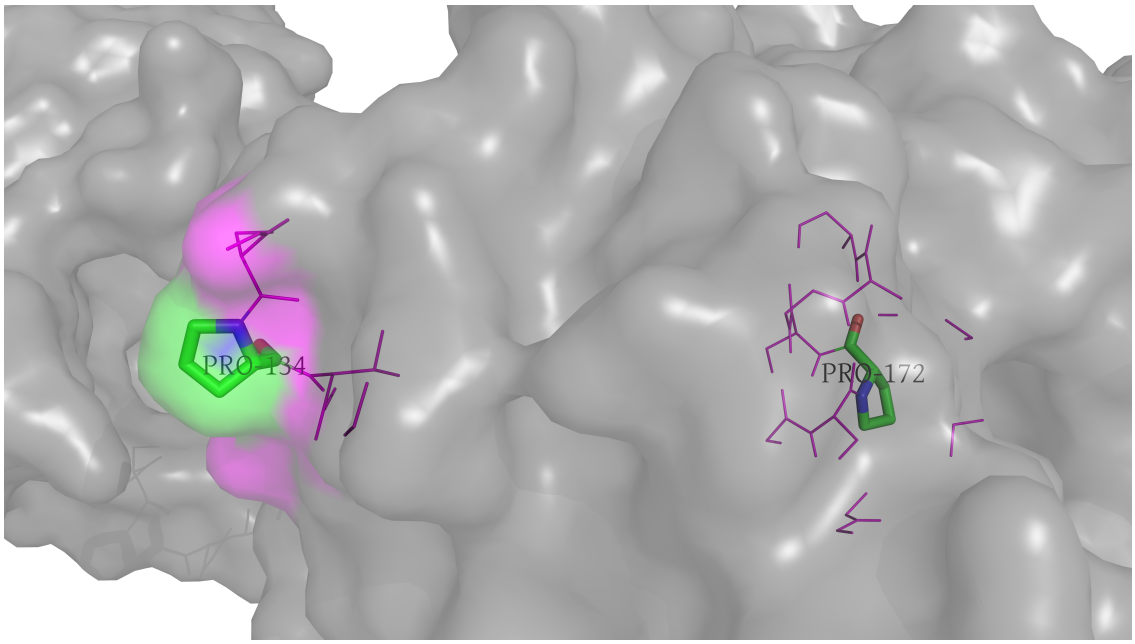

Figure 2.1: Proline 134 and 172 in *Escherichia coli* HSP70 chaperone: 5 Å environment comparison – NMR, PDB-ID 2KHO

$$H_{env} = (\sum_{i=1}^{n} H(aa_i))\rho \tag{2.2}$$

Polarity The scale defined by GRANTHAM in 1974 was used for the abstraction of the environment polarity [Grantham, 1974]. Table 2.5 contains the values assigned to each amino acid. A high value means that the amino acid is strongly polar, a low value indicates less polar amino acids. Equation 2.3 shows how the polarity of the environment (P_{env}) around proline containing n amino acids is abstracted, where $P(aa_i)$ is the numeric value for the amino acid aa_i.

Table 2.5: Polarity scale by GRANTHAM (1974)

aa	**Ala**	**Arg**	**Asn**	**Asp**	**Cys**	**Gln**	**Glu**	**Gly**	**His**	**Ile**
P(aa)	8.1	10.5	11.6	13.0	5.5	10.5	12.3	9.0	10.4	5.2
aa	**Leu**	**Lys**	**Met**	**Phe**	**Pro**	**Ser**	**Thr**	**Trp**	**Tyr**	**Val**
P(aa)	4.9	11.3	5.7	5.2	8.0	9.2	8.6	5.4	6.2	5.9

$$P_{env} = (\sum_{i=1}^{n} P(aa_i))\rho \tag{2.3}$$

Mutability The relative mutability scale of amino acids was defined by DAYHOFF ET AL. in 1978 and used to describe the relative mutability of the proline surrounding environ-

ment [Margaret O. Dayhoff, 1978]. Table 2.6 shows the numeric values for each amino acid. In respect to alanine ($M(Ala) = 100$) DAYHOFF ET AL. determined the probability of every amino acid for a mutation over an evolutionary period. Higher values indicate a more often observed mutation of this amino acid compared to alanine. Equation 2.4 shows how the mutability of the environment (M_{env}) around proline containing n amino acids is abstracted, where $M(aa_i)$ is the numeric value for the amino acid aa_i.

Table 2.6: Relative mutability scale by DAYHOFF ET AL. (1978)

aa	**Ala**	**Arg**	**Asn**	**Asp**	**Cys**	**Gln**	**Glu**	**Gly**	**His**	**Ile**
M(aa)	100	65	134	106	20	93	102	49	66	96
aa	**Leu**	**Lys**	**Met**	**Phe**	**Pro**	**Ser**	**Thr**	**Trp**	**Tyr**	**Val**
M(aa)	40	56	94	41	56	120	97	18	41	74

$$M_{env} = (\sum_{i=1}^{n} M(aa_i))\rho \tag{2.4}$$

Bulkiness The bulkiness of the environment was determined using the scale by ZIMMERMANN ET AL. (1968). It defines how bulky an amino acid residue is [Zimmerman et al., 1968]. The values in table 2.7 were assigned to each amino acid. A high value means the amino acid residue is bulky. The environment bulkiness (B_{env}) was calculated as seen in equation 2.5. The environment contains n amino acids, where $B(aa_i)$ is the numeric value for the amino acid aa_i.

Table 2.7: Bulkiness scale by ZIMMERMANN ET AL. (1968)

aa	**Ala**	**Arg**	**Asn**	**Asp**	**Cys**	**Gln**	**Glu**	**Gly**	**His**	**Ile**
B(aa)	11.50	14.28	12.82	11.68	13.46	14.45	13.57	3.40	13.69	21.40
aa	**Leu**	**Lys**	**Met**	**Phe**	**Pro**	**Ser**	**Thr**	**Trp**	**Tyr**	**Val**
B(aa)	21.40	15.71	16.25	19.80	17.43	9.47	15.77	21.67	18.03	21.57

$$B_{env} = (\sum_{i=1}^{n} B(aa_i))\rho \tag{2.5}$$

Flexibility The amino acid flexibility is described by the conformation of a residue. Larger residues are often less flexible due to steric constraints. The flexibility of amino acids was defined by BHASKARAN and PONNUSWAMY in 1988 [Bhaskaran and Ponnuswamy, 1988]. The values seen in table 2.8 were determined for each amino acid. The higher the value, the more flexible the amino acid is. The most flexible amino acid is the simplest amino acid glycine. The environment flexibility (F_{env}) was calculated as seen in equation 2.6. Where n amino acids is the number of amino acids in the environment and $F(aa_i)$ is the numeric value of flexibility for the amino acid aa_i.

Table 2.8: Flexibility scale by BHASKARAN and PONNUSWAMY (1988)

aa	**Ala**	**Arg**	**Asn**	**Asp**	**Cys**	**Gln**	**Glu**	**Gly**	**His**	**Ile**
F(aa)	0.36	0.53	0.46	0.51	0.35	0.49	0.50	0.54	0.32	0.46
aa	**Leu**	**Lys**	**Met**	**Phe**	**Pro**	**Ser**	**Thr**	**Trp**	**Tyr**	**Val**
F(aa)	0.37	0.47	0.30	0.31	0.51	0.51	0.44	0.31	0.42	0.39

$$F_{env} = (\sum_{i=1}^{n} F(aa_i))\rho \tag{2.6}$$

2.3.4 Proline Energy Approximation

The energy of an amino acid can be approximately determined by using a coarse-grained energy model based on inside/outside (see subsection 2.3.2) statistics of amino acids. The energy for an amino acid type can then be calculated using equation 2.7 [Labudde et al., 2012].

$$E(aa) = -\ln(\frac{n_{aa}^{in}}{n_{aa}^{out}}) \tag{2.7}$$

Where n_{aa}^{in} is the total occurrence of an amino acid buried in the protein structure (*inside*) and n_{aa}^{out} is the total occurrence of an amino acid on the protein surface (*outside*).

The energy statistics for 2700 non-redundant globular proteins found on the Energy Profile Suite (ePros) server[7] were used (table 2.9) to approximately calculate the energy of proline.

Table 2.9: Average coarse energy according to amino acid, dataset of 2700 non-redundant proteins

aa	**Ala**	**Arg**	**Asn**	**Asp**	**Cys**	**Gln**	**Glu**	**Gly**	**His**	**Ile**
E(aa)	-14.71	-5.94	-4.33	-1.83	-26.03	-5.30	-0.88	-7.62	-12.60	-27.33
aa	**Leu**	**Lys**	**Met**	**Phe**	**Pro**	**Ser**	**Thr**	**Trp**	**Tyr**	**Val**
E(aa)	-25.33	-0.74	-24.21	-25.68	-5.45	-6.93	-8.40	-20.49	-19.64	-22.6

To determine the energy of proline the energy of all amino acids within an 8 Å environment around proline was summarized (equation 2.8).

$$e_{Pro} = \sum_{i=1}^{n} E(aa) \tag{2.8}$$

[7] http://bioservices.hs-mittweida.de/

Table 2.10: Overview of the found prolines and the prolines in the training data set

	total	**relative**	***cis***	***trans***	***cis*/*trans***	**unknown**
prolines in dataset	71514	100 %	3195	67801	205	313
prolines in training dataset	69771	97.56 %	3141	66630	-	-

2.4 Creation of Training Data

With the help of the developed software Xaa-PIPT, which is discussed in subsection 2.10.1 and the previously described methods for structural feature extraction, a training dataset for the SVM was generated. It can be found on the enclosed CD (content see appendix A).

Every proline is representing one training instance. The isomerization state of Xaa-Pro is used as class label for the SVM. For the *cis* conformation `-1` was assigned as class label and for the *trans* conformation `1`, respectively. The *cis*/*trans* intermediate state was excluded from the training data. The SVM can only handle two-class problems unless a multi-class SVM is used.

Not every protein structure could be processed. Some chains did not contain any prolines or were invalid for other reasons. Likewise, as mentioned before, it is not possible to extract all features for every proline that was found. There were even prolines where it was not possible to calculate ω. This happened for example for prolines at the beginning of a chain, where no preceding amino acid Xaa exists. If one feature was not defined, the proline had to be excluded from the training data set. Table 2.10 shows how much information was lost due to undefined features. The procedure yielded 69771 prolines (3141 in *cis* and 66630 in *trans* conformation), excluding the intermediate isomerization state and any invalid proline or prolines with undefined features.

The whole training dataset contained much too many instances in respect to a reasonable computational training time. Beside this there was a huge imbalance between the two classes, which would have resulted in a trained classifier preferring the overrepresented class (*trans*). There are two possibilities to solve these problems. The first would be to apply a weight parameter w to the underrepresented class (*cis*) to avoid domination of the larger class. But this would not solve the problem of too many training vectors. So it was decided to create subsets of the original dataset to reduce the number of training vectors per data set and save computational time. Training time does not grow linearly and is significant higher for training data with a large number of training vectors. The following steps were performed ten times to create ten representative subsets:

1. select randomly 70 % of the underrepresented *cis* class
2. add the equal number of the overrepresented *trans* class randomly

Using this method ten representative and scaled subsets of equal size (4397 training vectors) and balanced classes (2199 *cis* and 2198 *trans*) were produced. The partial and random selection statistically ensures that a large variety of possible combinations between the two classes emerges. The resulting ten subsets were then used for the training of the classifier and as testing data. Namely the following ten subsets were created:

- `svm_balanced_1.pipt`
- `svm_balanced_2.pipt`
- `svm_balanced_3.pipt`
- `svm_balanced_4.pipt`
- `svm_balanced_5.pipt`
- `svm_balanced_6.pipt`
- `svm_balanced_7.pipt`
- `svm_balanced_8.pipt`
- `svm_balanced_9.pipt`
- `svm_balanced_10.pipt`

2.5 Scaling Features

Scaling features before applying a SVM is a very important step. The major effect of scaling is to avoid large numbers to dominate those in smaller numeric ranges. Beside this scaling avoids numerical difficulties for the SVM during calculation. Scaling is recommended between a range of $[-1;1]$. Furthermore the scaling for the training and the testing data must be equivalent [Chih-Wei et al., 2010]. To ensure correct scaling each feature was scaled regarding its maximal and minimal value. The same scaling was applied for training and testing data. Equation 2.9 was used for scaling each feature to the range of $[-1;1]$.

$$x_{scaled} = \frac{2(x - x_{min})}{x_{max} - x_{min}} - 1 \tag{2.9}$$

Where x is the unscaled feature, x_{min} the minimum of the feature type over all data sets and x_{max} the maximum, respectively. The scaling parameters were stored and then used for scaling of unknown test data by μXaa-PIPT.

2.6 Support Vector Machine Feature Selection

There are several possibilities known to estimate the weight of the used features for the SVM. F-score selection is a very simple and quite effective method to determine the

importance of features. A disadvantage of F-score selection is, that it does not reveal mutual information about features [Chen and Lin, 2006].

A preliminary python script called `fselect.py` was used to analyze the importance of features. This tool is included in the libSVM package. The output shows the importance of each feature and the corresponding testing results to estimate the prediction accuracy.

2.7 Support Vector Machine Parameter Grid-Search

As already mentioned in section 1.4, there is a penalty parameter C needed for the training of the SVM. Beside this every kernel has specific parameters. All of these parameters are dependent of the training data and thus unknown, but essential for the quality of the classifier. So it is necessary to determine the parameter C and the parameter for the used kernel (e. g. γ in case of RBF kernel) [Chih-Wei et al., 2010].

Cross-validation Using the n-fold cross-validation technique is a good approach to determine the power of a classifier to predict unknown test data. The dataset is divided in n subsets of equal size. After that one subset is tested against the classifier which was trained on the remaining $n-1$ subsets. This is done for each subset so that the whole training set is predicted. The cross-validation can help to prevent the classifier from overfitting the classes.

The cross-validation approach can be used to determine the parameters for the SVM. Various pairs of parameters are tried and the pair with the best cross-validation accuracy is picked. It was found out that exponential growing sequences of parameters (C: $[2^{-5};2^{15}]$ and γ: $[2^{-15};2^{-3}]$) are a well established and suitable practical method to identify good parameters [Chih-Wei et al., 2010].

A five-fold cross-validation grid-search for each kernel was performed using the python script `grid.py`, which is included in the libSVM package.

2.8 Support Vector Machine Training

After the correct parameters ($C = 2^{-1}$ for linear kernel, $C = 2^5$ for polynomial kernel, $C = 2^{15}$ and $\gamma = 2^{-3}$ for RBF kernel, see section 3.4) were found, the SVM was trained with each of the ten sub datasets using different kernel functions. The degree of the polynomial kernel function was set to $d = 3$. The resulting ten model files were then used for the evaluation of the prediction accuracy.

CUDA acceleration Due to very long training time for a large number of training vectors, acceleration by the graphics processing unit (GPU) was used. The CUDA reimplementation of libSVM 3.0 provided by MKLAB[8] was compiled for the use with a Nvidia GTX460 graphics card. The grid-search and the SVM training using the GPU resulted in much lower computing time.

2.9 Support Vector Machine Prediction

To perform a representative prediction, each of the ten models was predicted against the nine remaining sub datasets, which were not used for training of the model. Thus a ten-fold cross-validation was performed to find the best classifier. The software μXaa-PIPT uses the model file with the best prediction performance to predict unknown test data.

The accuracy (ACC) of each model file was calculated as seen in equation 2.10.

$$ACC = \frac{TP + TN}{n} \tag{2.10}$$

Likewise the Matthews correlation coefficient (MCC) (equation 2.11), the sensitivity or true positive rate (TPR) and the specificity or true negative rate (TNR) were also calculated using the equations 2.12 and 2.13. The ROCR[9] package for R was used for the automation of the calculations.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{2.11}$$

$$TPR = \frac{TP}{n_{trans}} \tag{2.12}$$

$$TNR = \frac{TN}{n_{cis}} \tag{2.13}$$

Where n (n_{cis}, n_{trans}) is the number of (*cis* or *trans*) samples, TP is the number of true positive predictions, TN of true negatives, FP of false positives and FN is the number of false negative predictions, resulting from a binary confusion matrix [Sing et al., 2005]. The negatives were defined as *cis* and the positive values as *trans*, leading to the confusion matrix seen in table 2.11.

[8] http://mklab.iti.gr/project/GPU-LIBSVM
[9] http://rocr.bioinf.mpi-sb.mpg.de/

Table 2.11: Confusion matrix example

	predicted	
	cis	*trans*
cis	TN	FP
trans	FN	TP

The MCC yields a number between -1 and 1. A value of 1 indicates a perfect prediction and a value of 0 indicates a random prediction. Values less then 0 are indicating a even worse than random prediction [Sing et al., 2005]. The TPR (sensivity) relates to the classifiers ability to identify *trans* prolines, whereas the TNR (specificity) is related to the correct identification of *cis* prolines. A TPR and a TNR of 1 would describe a perfect classifier.

Beside this the so called receiver operating characteristics curve (ROC) provides a good graphical representation of the TP and the FP prediction rate of a classifier. ROC curves are obtained by plotting the 1-specificity or false positive rate (FPR) against the sensitivity or TPR [Song et al., 2006]. The area under the ROC curve (AUC) is equal to the value of the Wilcoxon-Mann-Whitney test – a non-parametric statistical test – and shows the probability that a classifier will score a randomly drawn positive sample higher than a randomly negative drawn, respectively [Sing et al., 2005]. The higher the AUC value is, ranging from 0 to 1, the better the classifier performs.

2.10 Implementation in Java

To perform structural feature extraction and all SVM related steps the software Xaa-PIPT was developed. Based on the calculations and the results achieved with Xaa-PIPT, the project μXaa-PIPT was forked. This lightweight software was designed for the use without any previous knowledge about machine learning tools to ease the usage. It is addressed to anyone who wants to investigate the Xaa-Pro isomerization in proteins, either available from the PDB, or locally provided in PDB structure format.

Xaa-PIPT and μXaa-PIPT can be found on the attached CD. Beside the binary files the source code is included. An overview of the CD content is shown in appendix A.

2.10.1 Xaa-PIPT

Requirements There were several basic requirements to met for the calculation software Xaa-PIPT. The following list summarizes the requirements:

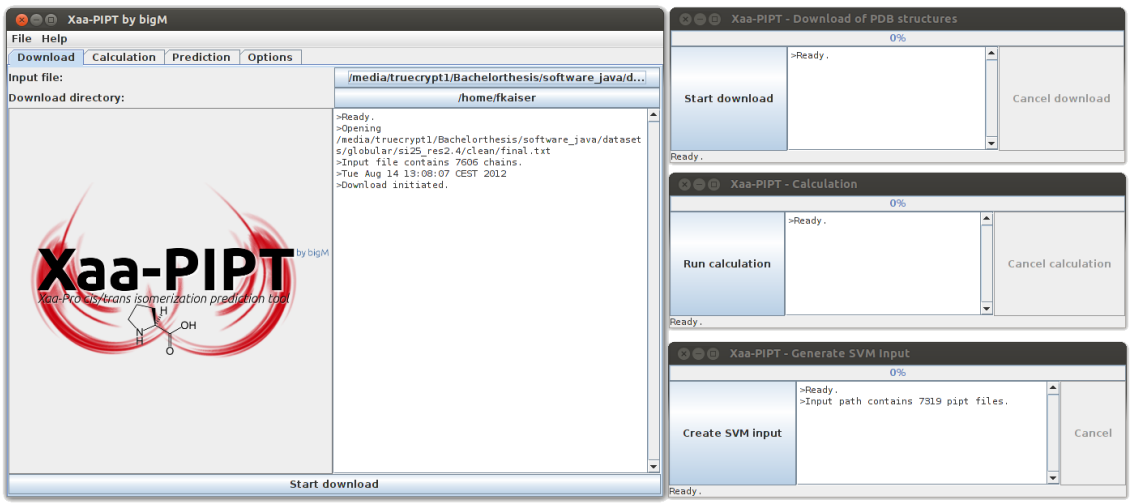

Figure 2.2: Xaa-PIPT GUI layout

- process large data amounts
- multithreading capability
- extract structural features
- highly configurable parameters
- SVM implementation
- store results for every chain

The software was developed concerning these requirements.

Graphical user interface Beside the basic requirements it was necessary to implemented a graphical user interface. Figure 2.2 shows the basic layout of the software. Further screenshots can be found in appendix B.

Functions To realize the requirements the following functions were implemented:

- The ability to bulk download the PDB structure files of the dataset for local storage.
- Calculation of the ω angle and extraction of sequence fragments up to three positions before and after proline.
- Feature extraction of secondary structure information, inside/outside classification of proline, environment hydrophobicity, polarity, mutability, bulkiness, flexibility and energy approximation of proline.
- Conversion of features into libSVM format, scaling and random generation of equal sized datasets and training of SVM together with prediction functionality.
- Selection of input file format, hydrophobicity and polarity scales, energy calculation type and the definition of the environment radius. Beside this the intervals for the *cis* and *trans* conformation can be set individually.

File formats There are two possible input file formats for Xaa-PIPT. Firstly it accepts the standard output received from PISCES in the following format:

```
IDs         length   Exptl.  resolution   R-factor FreeRvalue
7ODCA       424      XRAY        1.600    0.20     0.23
1GH9A        71      NMR            NA    1.00     1.00
2QUDA       125      XRAY        1.600    0.16     0.20
...
```

Secondly an own list of protein chains can be used in the following format (`PDB-ID_CHAIN-ID`):

```
1A0S_P
2A06_E
2A06_G
...
```

To store the results for each chain a new file format, the `.pipt` format, was introduced. It is based on the idea of the comma-seperated values (CSV) format. The following example shows an extraction of the `.pipt` file for chain A of PDB-ID 1T3T (the header is shown in the first line):

```
res;omega;cis/trans;xPx;xxPxx;xxxPxxx;ss_i-2;ss_i-1;ss;ss_i+1;ss_i+2; ...
-4;179.9318;trans;VPR;LVPRG;GLVPRGS;s;s;s;s;c;outside;0.0122;0.1688;1.78; ...
9;179.5616;trans;SPA;GSPAL;RGSPALS;s;s;s;c;c;outside;-0.0191;0.2261;2.4599; ...
46;-179.7322;trans;APL;NAPLN;LNAPLND;c;c;c;c;h;outside;-0.0102;0.0662;0.713; ...
...
```

The libSVM format was used for the SVM input. The class label is printed first, followed by the feature number, colon and the feature value. An example is shown below.

```
...
 1.0 1:1.0000 2:1.0000 3:1.0000 4:-1.0000 5:-1.0000 6:1.0000 7:0.3110 ...
-1.0 1:1.0000 2:1.0000 3:-1.0000 4:-1.0000 5:-1.0000 6:-1.0000 7:0.4165 ...
...
```

Where for example feature number 3 represents the secondary structure of proline.

2.10.2 μXaa-PIPT

Requirements Based on the algorithms for feature extraction used in Xaa-PIPT, μXaa-PIPT was developed. The following requirements were fulfilled:

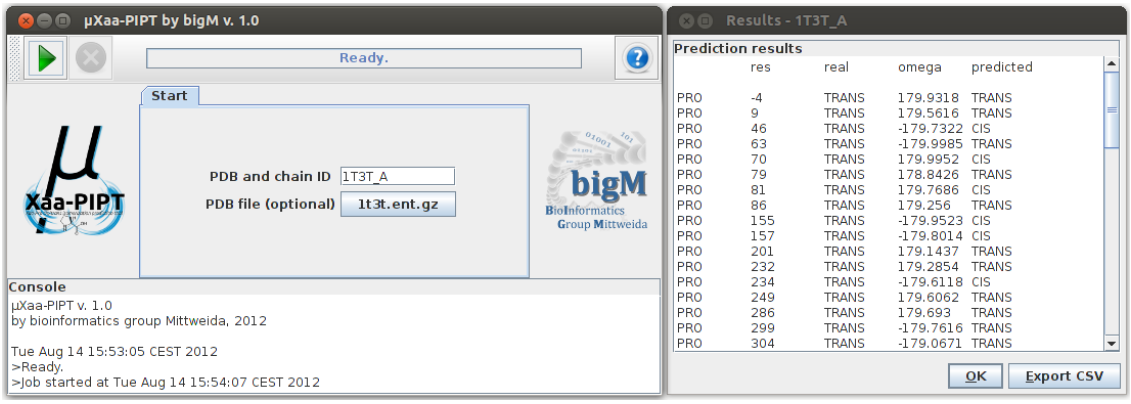

Figure 2.3: μXaa-PIPT GUI layout

- multithreading capability
- standalone solution
- no parameter adjustment necessary
- process each proline separately

There are two files necessary for the software: a SVM model file and a file containing the feature scaling parameters of the used training data. These two files are included in the classpath and can be substituted only by a developer and with the help of Xaa-PIPT for the creation of the two files.

Graphical user interface The GUI of μXaa-PIPT is shown in figure 2.3. It was kept as simple as possible and self-explanatory. There are only two basic elements: the main window to run prediction jobs and the results window, which is also used for the output of errors.

Functions and functionality The following functions were implemented in μXaa-PIPT:

- accept input in the format `PDB-ID_CHAIN-ID` (e. g. `1T3T_A`)
- download structure automatically or specify a local file
- extract features for each proline
- predict isomerization state based on the included SVM model
- export results in CSV format

If a PDB- and a chain ID is specified and a job is started, μXaa-PIPT calculates the features for each proline separately. After the features are calculated, the proline isomerization state is predicted. This is repeated for every proline found in the chain. After iteration over the complete chain an output is generated. It shows the residue number,

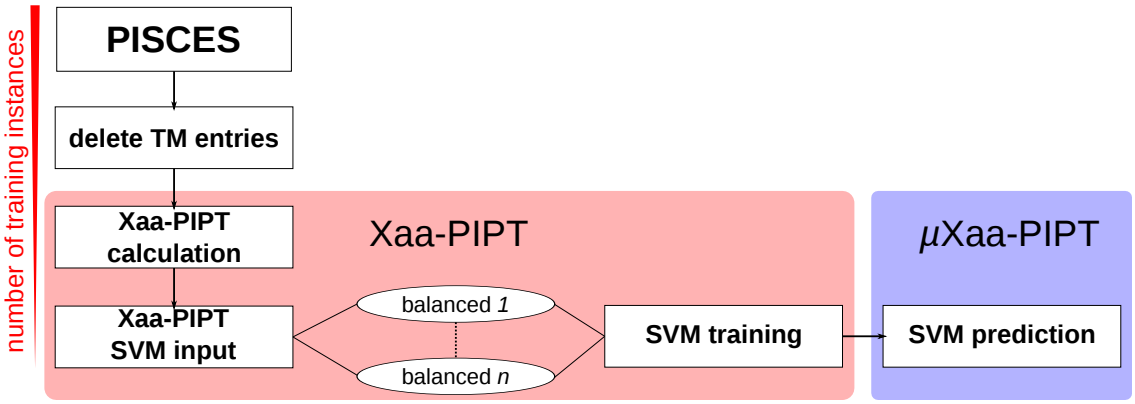

Figure 2.4: Methods overview, workflow diagram

the real isomerization state (calculated from the structure), the corresponding ω angle and the predicted isomerization state depending on the proline surrounding environment.

2.11 Methods Overview

To give a short overview of the applied methods a schematic workflow diagram is shown in figure 2.4.

The main working steps are showed as white boxes. As indicated in red on the left side of the diagram, the information (the number if training instances/vectors) decreases until the SVM training data has been created (mentioned in sections 2.2 and 2.4). Boxes shaded in red include all steps carried out with the help of Xaa-PIPT. For the prediction of unknown test data μXaa-PIPT can be used, even though Xaa-PIPT also implements a prediction functionality. In general, the following working steps succession was a topic of this work:

1. creation of a representative dataset using PISCES
2. deletion of transmembrane proteins
3. structural feature calculation with Xaa-PIPT
4. creation of SVM training data with Xaa-PIPT
5. creation of scaled and balanced subsets of equal size with Xaa-PIPT
6. parameter grid-search
7. training of the SVM
8. prediction of unknown data with Xaa-PIPT or μXaa-PIPT

3 Results

3.1 Dataset Analysis

3.1.1 Quantities of Isomerization States

As a coarse overview figure 3.1 shows the occurrence of the isomerization states in the dataset. There were 71201 out of 71514 prolines (99.56 %) in the dataset where a calculation of the ω angle was possible. The omega angle of the remaining 313 prolines (0.58 %) could not be determined.

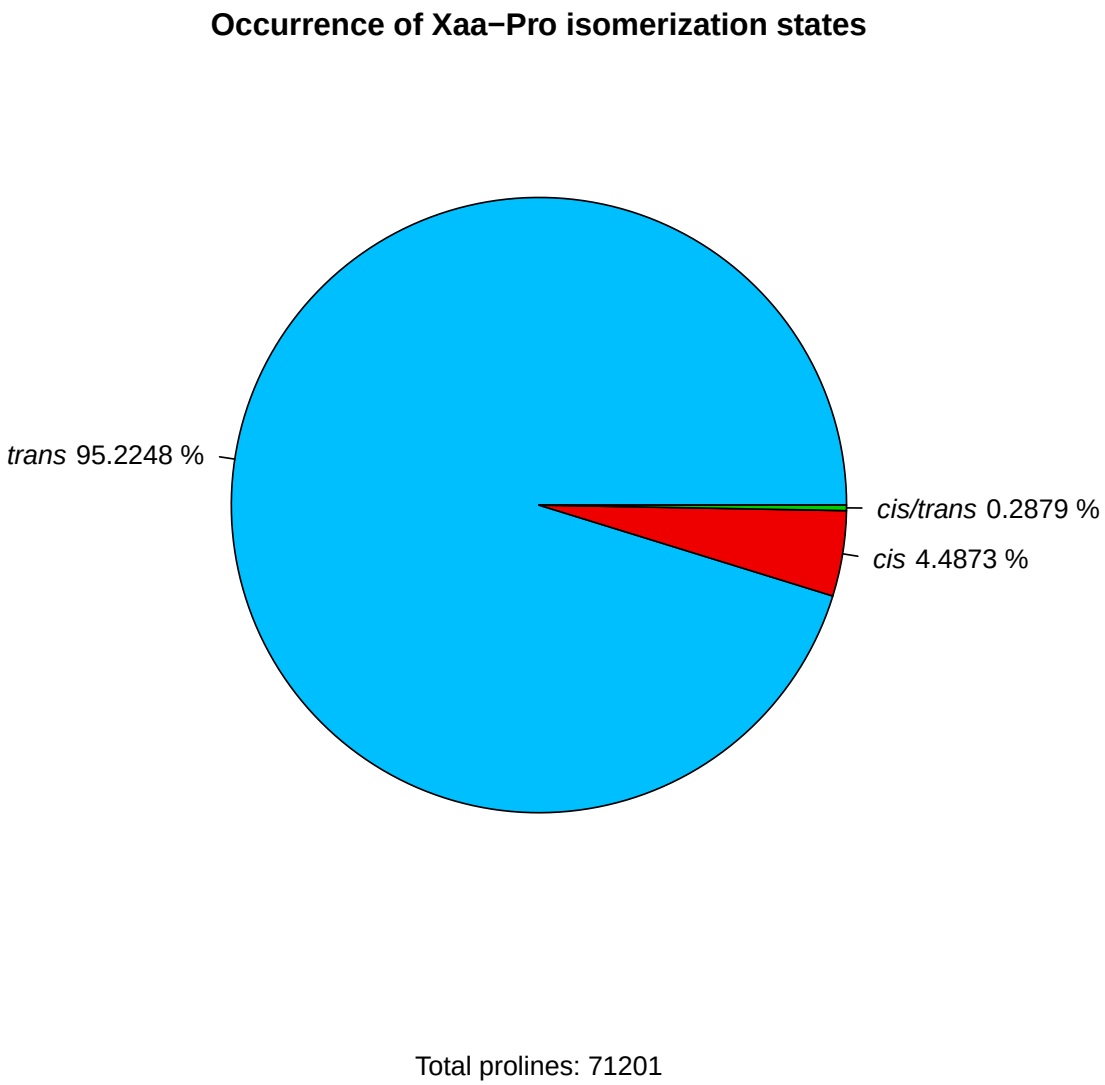

Figure 3.1: Quantities of proline isomerization states

3.1.2 Omega Angle Distribution

With the help of a boxplot it is possible to visualize the distribution of data without making any assumptions of how the data is distributed. Boxplots were used to show the distribution of the ω angle depending on the isomerization state of Xaa-Pro. In figure 3.2 the distribution of the *cis* conformation in 3195 occurrences is shown. Table 3.1 shows the corresponding values. The same procedure was performed for the *trans* isomerization state. The results are shown in figure 3.3 for 67801 occurrences and the corresponding values can be found in table 3.2.

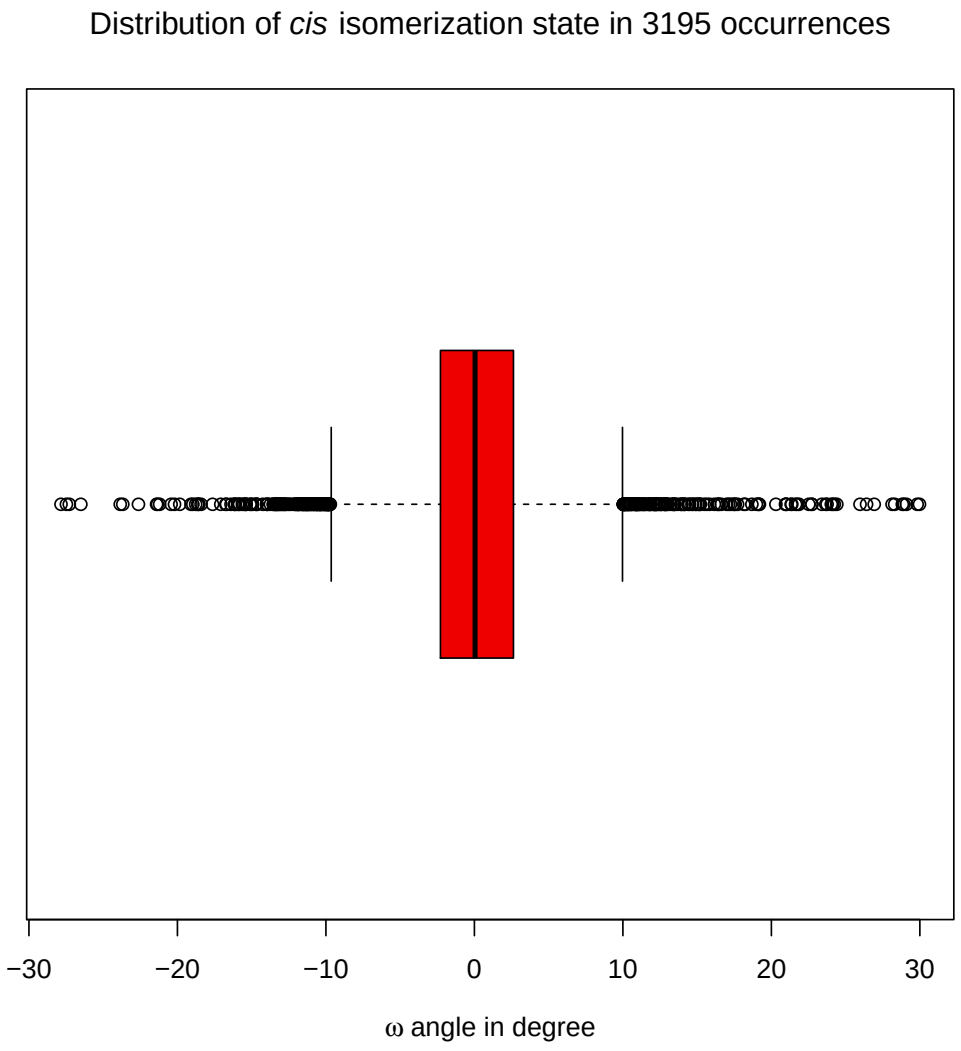

Figure 3.2: Boxplot for the distribution of the ω angle in *cis* isomerization state

Table 3.1: Distribution of the ω angle in *cis* isomerization state

minimum	**1st quartile**	**median**	**mean**	**3rd quartile**	**maximum**
-27.8400	-2.2870	0.0453	0.2159	2.6300	29.9700

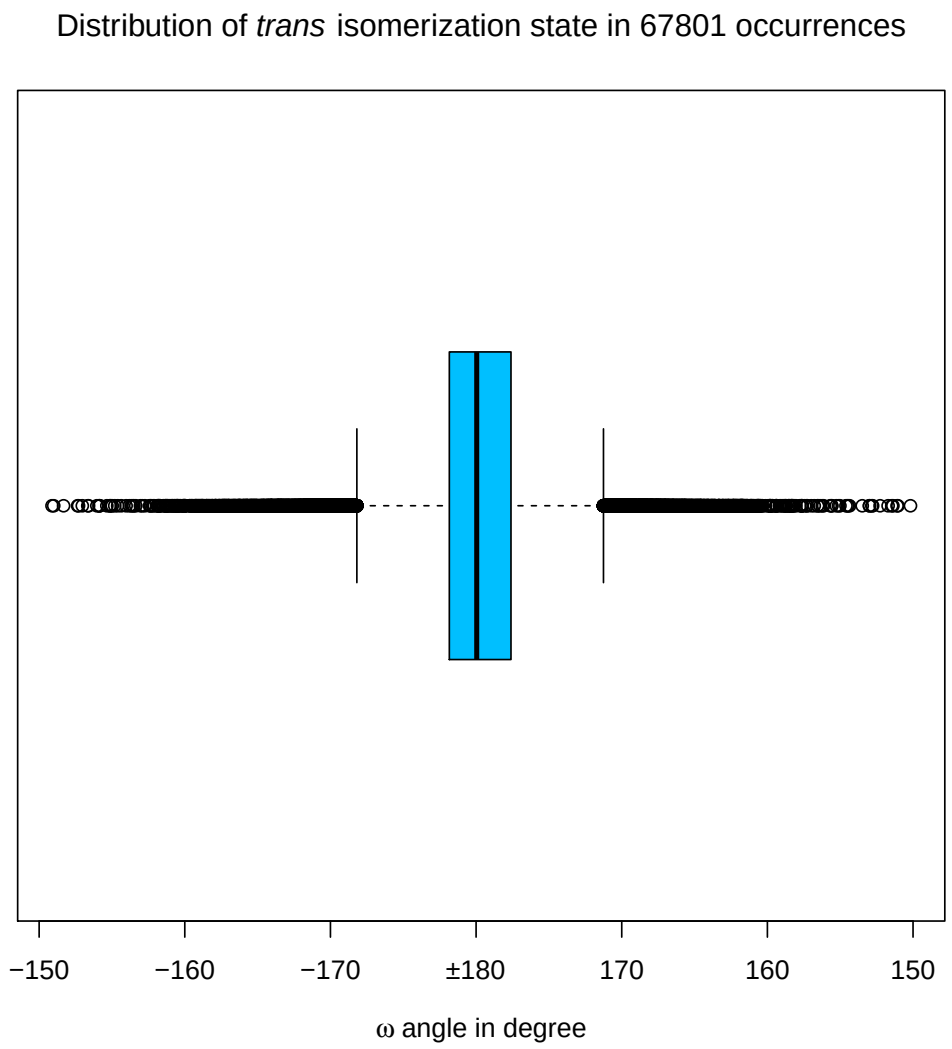

Figure 3.3: Boxplot for the distribution of the ω angle in *trans* isomerization state

Table 3.2: Distribution of the ω angle in *trans* isomerization state

minimum	**1st quartile**	**median**	**mean**	**3rd quartile**	**maximum**
-150.8800	-178.1640	179.9568	179.8263	177.6030	150.1800

3.2 Determination of Sphere Radius

To determine the ideal radius of the sphere defining the Xaa-Pro environment, a five-fold cross-validation loose grid-search was applied on a high-resolution dataset with 2236 protein chains. The PISCES criteria of this dataset are shown in table 3.3. Using the same procedure as for the large dataset, the high-resolution datasets for each radius were divided in ten subsets. After that a loose grid-search within C: $[2^{-5};2^{15}]$ and γ: $[2^{-15};2^{-3}]$ and a step size of 1 was performed using the RBF kernel. The average grid-search accuracy of all subsets was calculated to determine the best sphere radius. The results are shown in figure 3.4. The best accuracy of 64.2857 % (highlighted, red line) was reached with a sphere radius of 5 Å. A radius of 3 Å resulted in nearly no amino acids within the environment, which then produced extremely small sub datasets. This can considered to be a statistical outlier. The exact values of the sphere radius search are listed in table 3.4.

Table 3.3: PISCES criteria for the creation of the high-resolution dataset

criteria	**value**
sequence percentage identity	$\leq$ 20 %
sequence chain length	40 - 10000
resolution	0.0 - 1.0 Å
R-factor value	0.3
non-X-ray entries	include
C_α-only entries	exclude
cull PDB by	chain

Table 3.4: Five-fold cross-validation, average accuracy values depending on sphere radius

radius in Å	**average prediction accuracy**
3	81.2000
4	63.9872
5	**64.2857**
6	62.7505
7	62.7932
8	63.4542
9	63.3902
10	62.8785

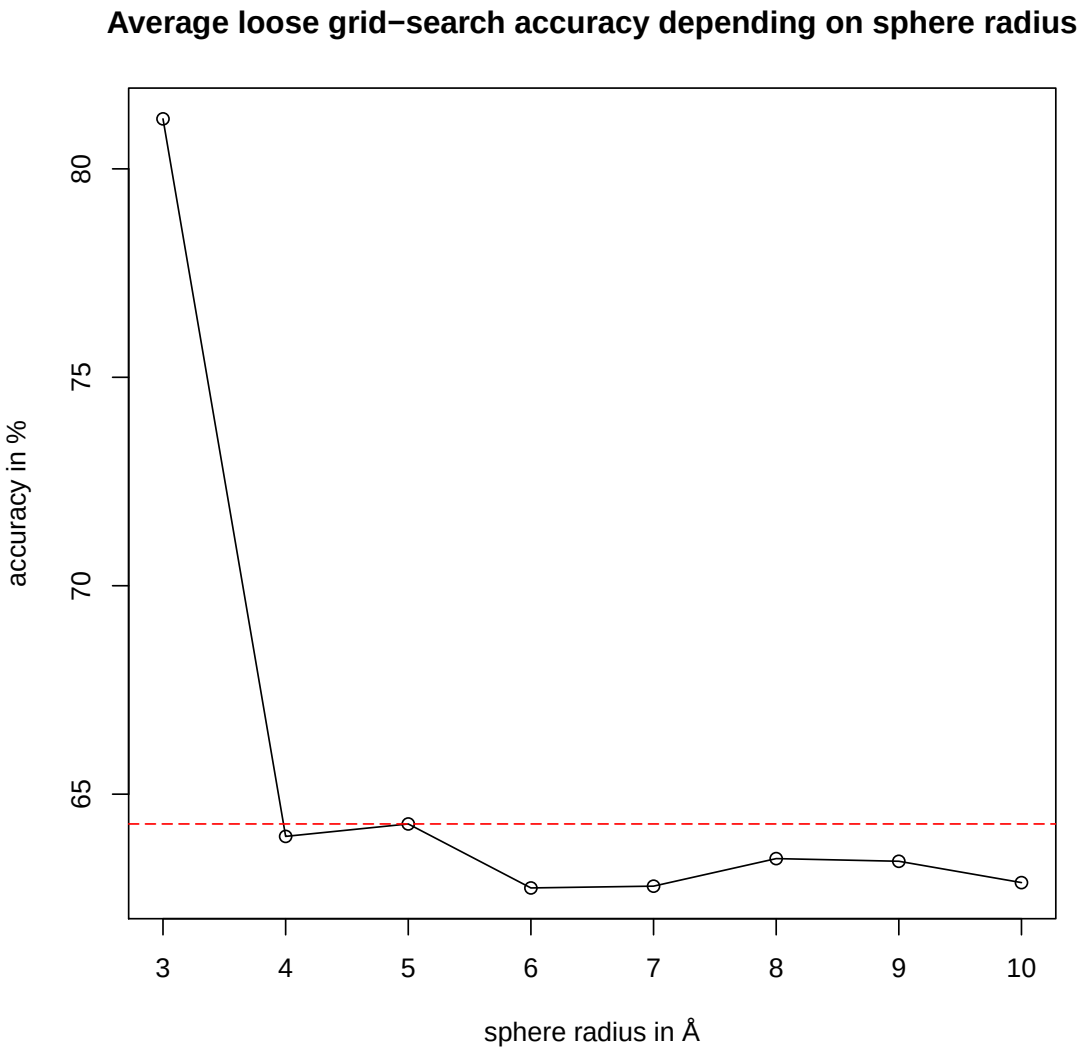

Figure 3.4: Five-fold cross validation accuracy depending on sphere radius

3.3 Feature Importance

Using the F-Score selection technique the feature importance was determined. The RBF kernel was used in combination with a five-fold cross-validation grid-search. Table 3.5 shows the results. The average prediction accuracy over all datasets was calculated in the last column. The highest F-score of 0.1027 was achieved by the secondary structure information of proline (highlighted value).

Table 3.5: F-score feature importance

feature	dataset 1	dataset 2	dataset 3	dataset 4	dataset 5	dataset 6	dataset 7	dataset 8	dataset 9	dataset 10	average
ss i-2	0.0224	0.0163	0.0213	0.0232	0.0208	0.0189	0.0193	0.0207	0.0251	0.0228	0.0211
ss i-1	0.0950	0.0809	0.0904	0.0899	0.0872	0.0774	0.0886	0.0763	0.0960	0.0897	0.0872
ss	0.1087	0.0923	0.1045	0.1116	0.1031	0.0912	0.1090	0.1003	0.0966	0.1093	**0.1027**
ss i+1	0.0493	0.0440	0.0524	0.0592	0.0482	0.0491	0.0590	0.0494	0.0518	0.0562	0.0519
ss i+2	0.0216	0.0175	0.0270	0.0277	0.0191	0.0216	0.0307	0.0202	0.0230	0.0210	0.0229
in/out	0.0032	0.0009	0.0007	0.0021	0.0011	0.0001	0.0013	0.0010	0.0011	0.0010	0.0012
H_{env}	0.0023	0.0015	0.0015	0.0049	0.0007	0.0024	0.0029	0.0032	0.0044	0.0018	0.0025
P_{env}	0.0097	0.0049	0.0049	0.0072	0.0047	0.0107	0.0029	0.0075	0.0095	0.0050	0.0067
M_{env}	0.0124	0.0076	0.0072	0.0085	0.0074	0.0139	0.0120	0.0106	0.0114	0.0098	0.0101
B_{env}	0.0115	0.0071	0.0082	0.0105	0.0078	0.0151	0.0102	0.0106	0.0125	0.0092	0.0103
F_{env}	0.0070	0.0038	0.0038	0.0059	0.0037	0.0092	0.0065	0.0058	0.0073	0.0034	0.0056
E_{Pro}	0.0104	0.0043	0.0060	0.0044	0.0068	0.0048	0.0089	0.0051	0.0048	0.0069	0.0062

3.4 Grid-Search Results

A loose five-fold cross-validation grid-search for C: $[2^{-5}; 2^{15}]$ in steps of 1 was done for the linear and the polynomial kernel function. And for the RBF kernel the grid-search was performed in steps of 1, using the intervals C: $[2^{-5}; 2^{15}]$ and γ: $[2^{-15}; 2^{-3}]$, respectively. A finer grid-search did not achieve higher prediction accuracy.

Figure 3.5 shows the results for each of the datasets. The red line indicates the average accuracy of the RBF kernel over all datasets (66.9525 %). The numeric results are shown in table 3.6.

Table 3.6: Loose grid-search accuracy values depending on kernel function

dataset	linear	polynomial	RBF
1	64.5668	66.7273	67.0457
2	63.0430	66.1133	66.6818
3	64.2484	66.1815	66.2952
4	64.4076	67.1594	67.8417
5	63.7025	66.6818	67.1139
6	63.3159	65.8858	66.4771
7	63.9982	67.2049	67.1367
8	63.5888	65.9541	66.0450
9	63.7480	66.7501	66.9775
10	64.4303	67.5688	**67.9099**
average	63.9050	66.6227	66.9525

The parameters of dataset 10 (RBF kernel) yielded the best accuracy of 67.9099 % (highlighted value). A detailed visualization of the RBF kernel grid-search performed on dataset 10 can be seen in figure 3.6.

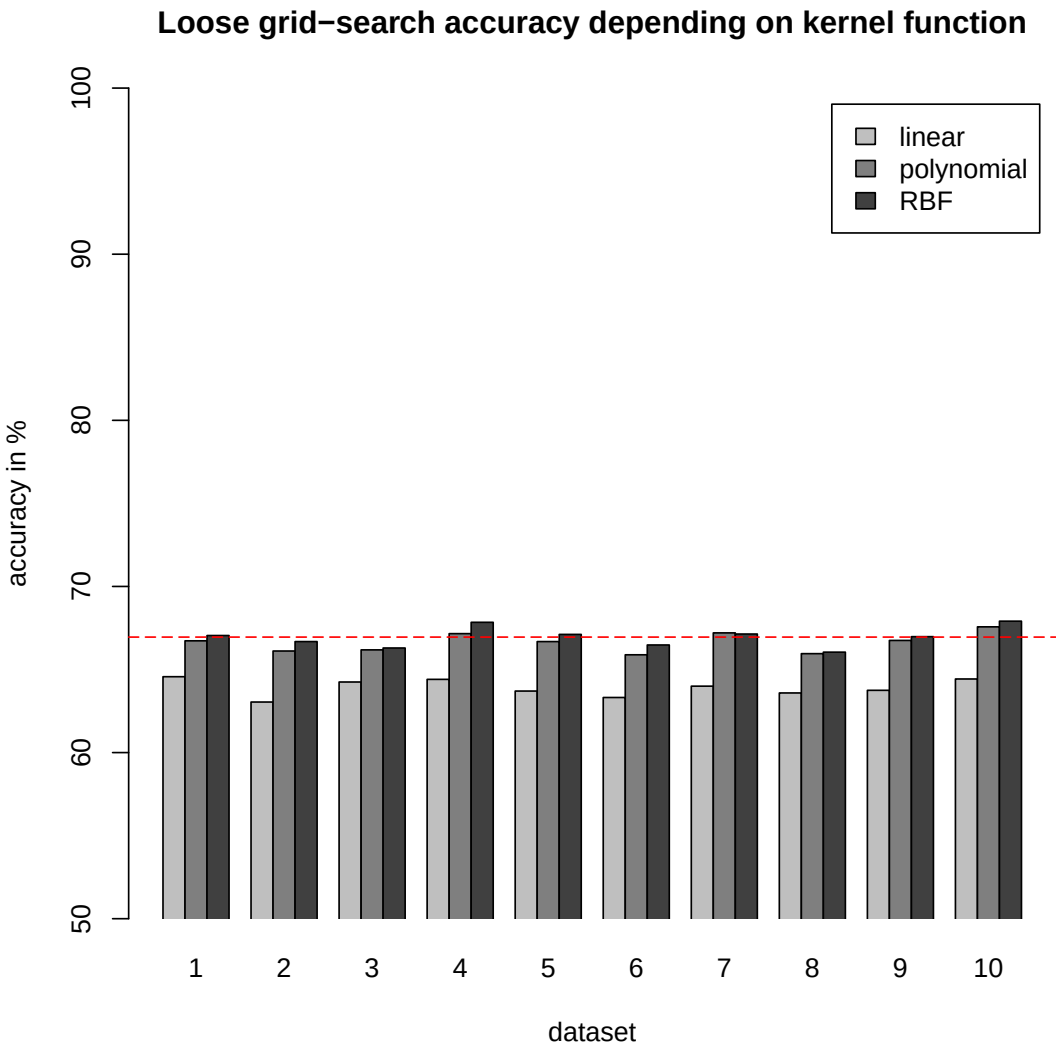

Figure 3.5: Loose grid-search results depending on kernel function

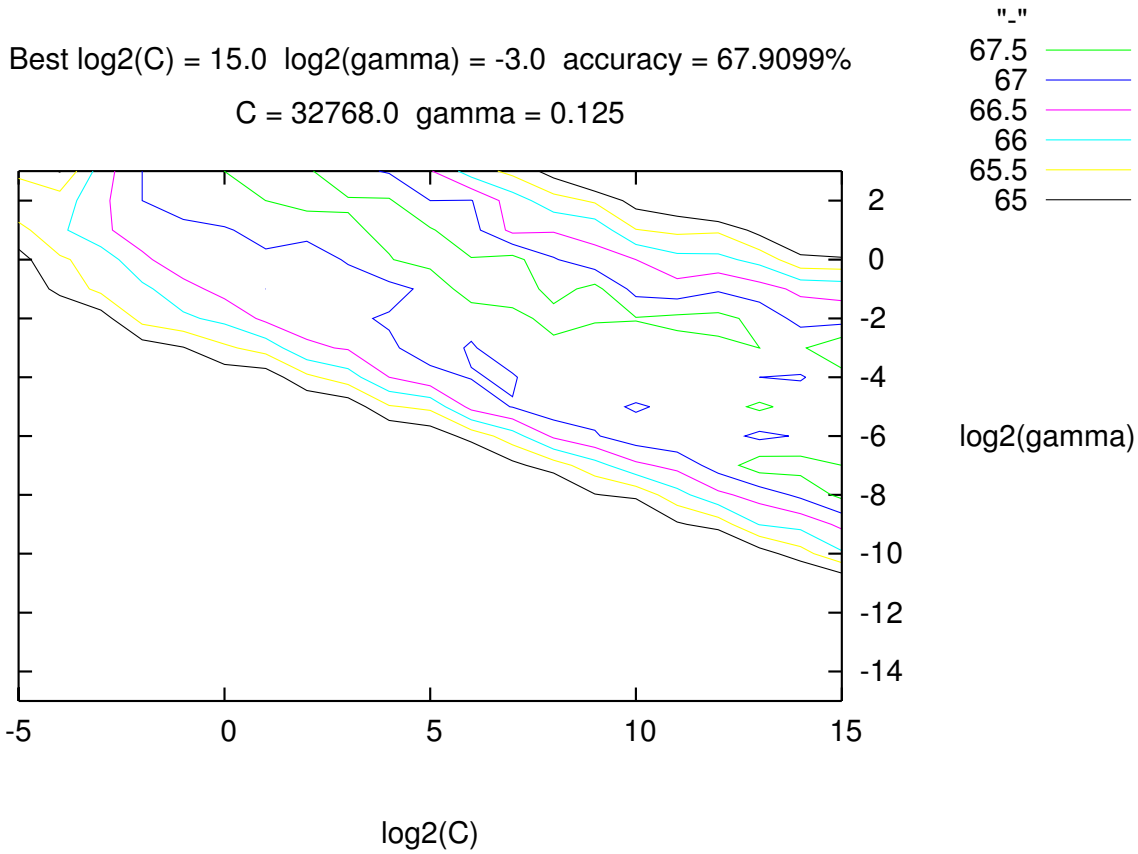

Figure 3.6: Loose grid-search results for dataset 10, RBF kernel

Therefore the following parameters of the RBF kernel were defined as the best and used for the training of the SVM:

$$C = 2^{15} \text{ and } \gamma = 2^{-3}.$$

Beside this the best C determined for the linear kernel was:

$$C = 2^{-1}.$$

And the polynomial kernel ($d = 3$) performed best with:

$$C = 2^{5}.$$

3.5 Prediction Performance

The ten model files, resulting from the RBF kernel training with the parameters $C = 2^{15}$ and $\gamma = 2^{-3}$, produced the prediction accuracy seen in table 3.7. This accuracy was determined using ten-fold cross-validation. Beside this the MCC, the sensitivity and the specificity for every dataset is shown. The best values are highlighted and an overall accuracy of all datasets is also shown in the last row. Model 8 performs best with a prediction accuracy of 70.0478 %, a MCC of 0.4223, a sensitivity of 0.5433 and a specificity of 0.8576. Therefore the model 8 was chosen for the integration in μXaa-PIPT.

Table 3.7: Prediction performance of the classifiers for RBF kernel training ($C = 2^{15}$ and $\gamma = 2^{-3}$), ten-fold cross-validation accuracy, MCC, sensitivity, specificity

dataset	accuracy	MCC	sensitivity	specificity
1	69.8405	0.4172	0.5437	0.8530
2	69.8658	0.4232	0.5262	0.8710
3	69.5424	0.4141	0.5300	0.8607
4	69.9492	0.4141	**0.5653**	0.8336
5	69.7597	0.4118	0.5569	0.8383
6	69.9467	0.4177	0.5509	0.8480
7	69.9492	0.4199	0.5435	0.8554
8	**70.0478**	0.4223	0.5433	0.8576
9	69.7723	**0.4235**	0.5185	**0.8769**
10	69.6283	0.4137	0.5384	0.8541
average	69.8302	0.4177	0.5417	0.8549

Figure 3.7 shows the ROC curve of the RBF kernel trained classifiers with the maximal AUC value (0.7004), achieved by model 8 and the minimal AUC value (0.6954), achieved by model 3. All AUC values can be seen in table 3.8, the highest AUC value is highlighted.

Table 3.8: AUC values of all RBF kernel trained classifiers

dataset	AUC
1	0.6984
2	0.6986
3	0.6954
4	0.6995
5	0.6976
6	0.6994
7	0.6995
8	**0.7004**
9	0.6977
10	0.6962
average	0.6983

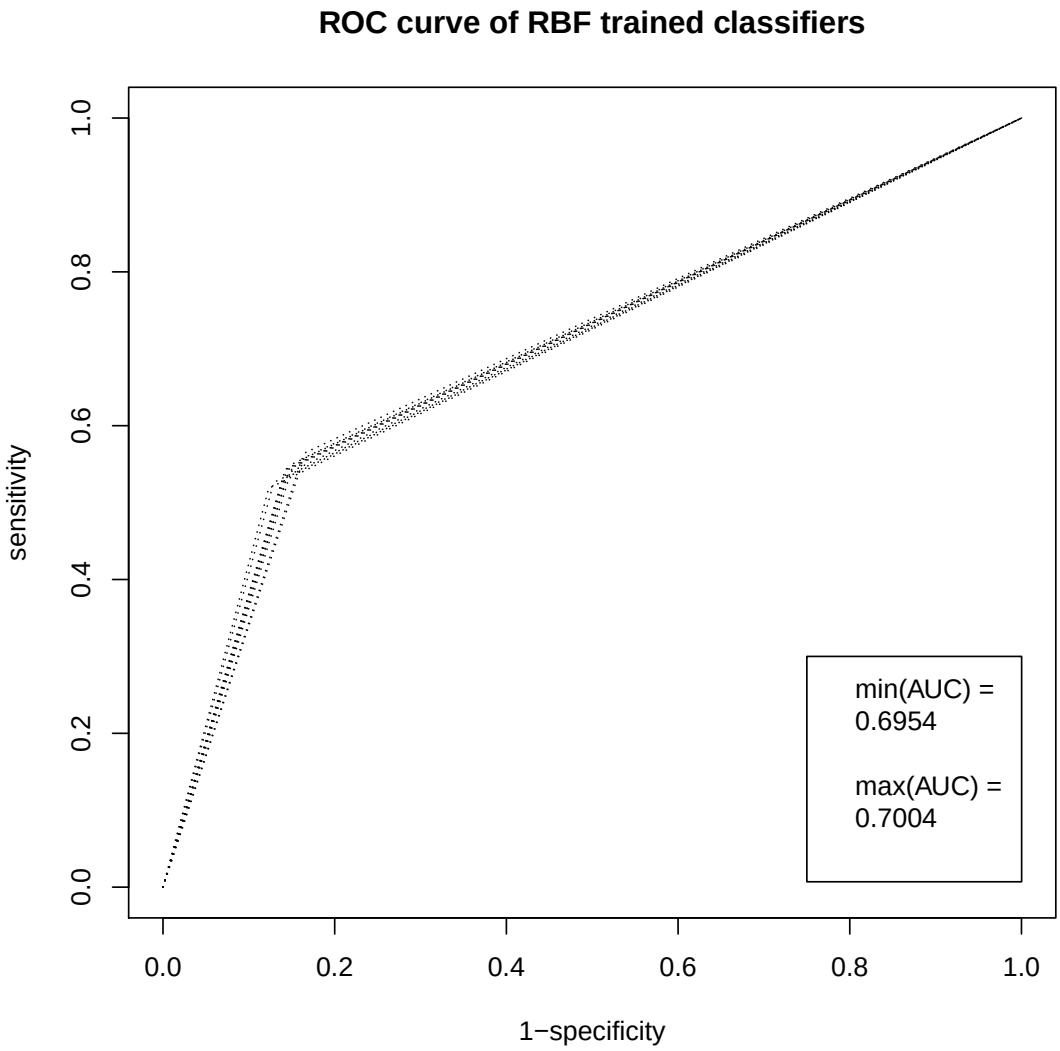

Figure 3.7: ROC curve of all RBF kernel trained classifiers

The linear ($C = 2^{-1}$) and the polynomial ($C = 2^{5}$) kernel functions were also used for the training of the classifiers to show a direct comparison to the RBF kernel. The results can be seen in figure 3.8 where the red line indicates the average prediction accuracy of the RBF kernel over all models (69.8302 %). The numeric results are shown in table 3.9.

Table 3.9: Ten-fold cross-validation accuracy values depending on kernel function

dataset	linear	polynomial	RBF
1	63.8693	68.1045	69.8405
2	63.4347	68.0666	69.8658
3	63.8895	68.1399	69.5424
4	63.9097	67.6143	69.9492
5	63.8744	67.9731	69.7597
6	63.2401	68.3446	69.9467
7	63.8693	68.2233	69.9492
8	63.2805	67.7634	**70.0478**
9	63.9679	68.0489	69.7723
10	63.8263	67.7255	69.6283
average	63.7162	68.0004	69.8302

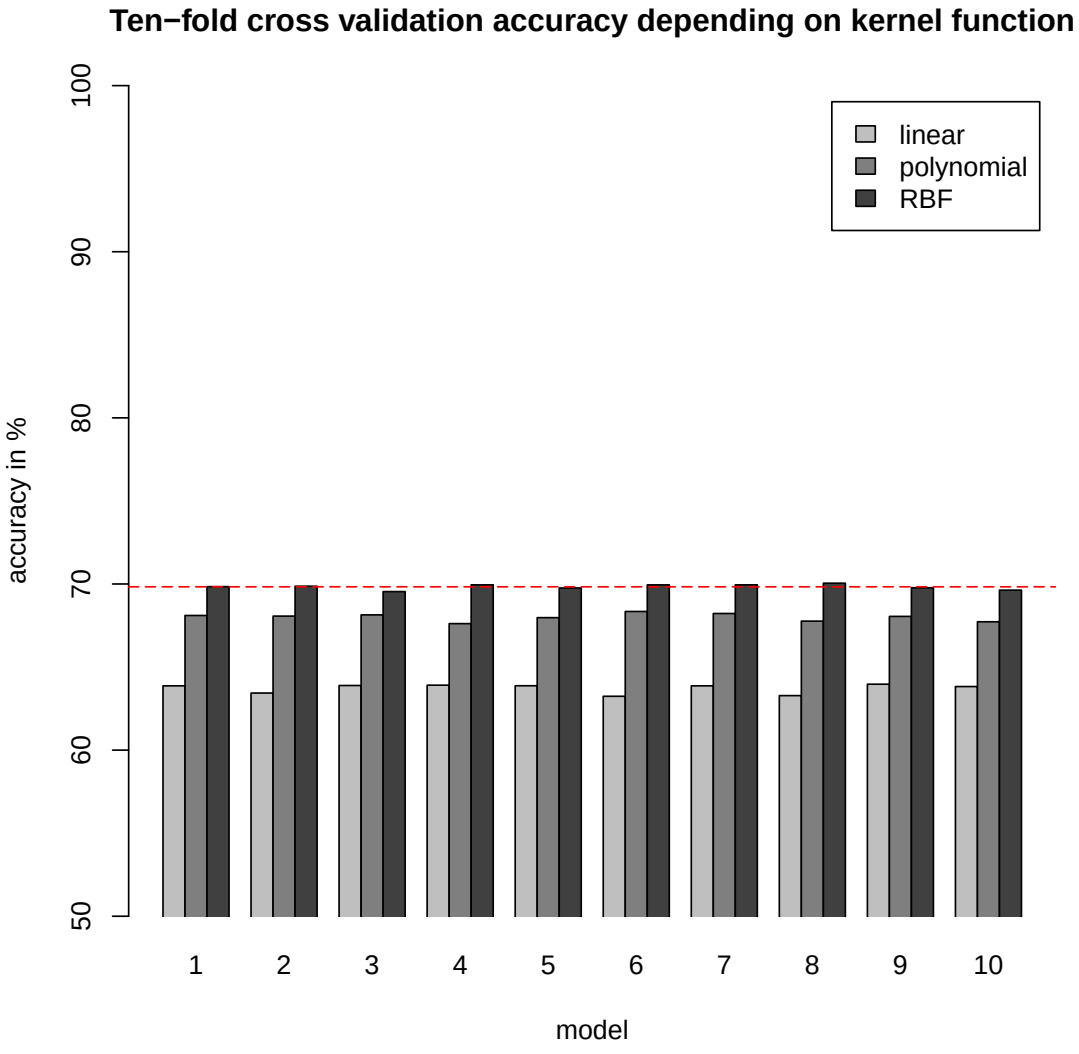

Figure 3.8: Ten-fold cross-validation accuracy depending on kernel function

3.6 Intermediate State Classification

Using the SVM model 8 all 205 occurrences of the *cis*/*trans* intermediate state in the dataset were predicted. Figure 3.9 shows a graphical overview of the prediction results. Table 3.10 shows the corresponding numeric values. For the unpredictable prolines it was not possible to extract all features necessary for the prediction. Remarkably 80.4020 % of the intermediate state prolines were predicted to be in *cis* isomerization state.

Table 3.10: Prediction quantities of 205 *cis*/*trans* intermediate state prolines

predicted	**absolute**	**relative**
cis	160	80.4020
trans	39	19.5980
unpredictable	6	2.9268

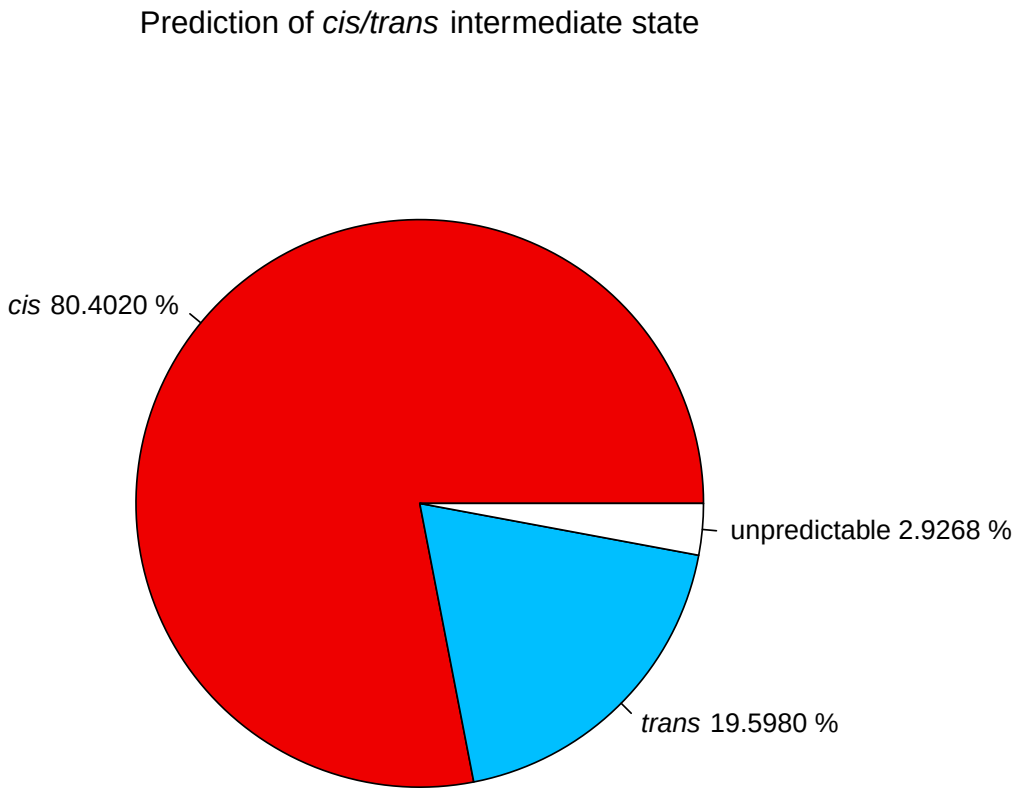

Figure 3.9: Prediction of 205 *cis*/*trans* intermediate state prolines

4 Discussion

4.1 Omega Angle Quantity and Distribution

The occurrences of the isomerization states in the used dataset correspond well with literature values. WEISS ET AL. proposed occurrences of *cis* Xaa-Pro between 4-5 % [Weiss et al., 1998] depending on the resolution of the structure. They gained a *cis* Xaa-Pro rate of 5.03 % in structures with a resolution between 2.0 Å and 2.5 Å. The difference to the results in this work can be affiliated to the *cis* interval, which was defined more generous by WEISS ET AL. ($-45° \leq \omega \leq 45°$).

The analysis of the distribution of the ω angle (section 3.1.2) showed, that the interval definitions for the isomerization are chosen quite well. The middle 50 % of the ω angles for the *cis* state distribute between -2.2870 ° and 2.6300 ° (see table 3.1) and there are only very few outliers below -20 ° and 20 °, respectively. The assumption can be made that there may be a more exact representation of the *cis* class if the interval is reduced to $-20° \leq \omega \leq 20°$. However, the effect on the learning and performance of the SVM would be very slightly, if it even exists. As $-30° \leq \omega \leq 30°$ is a commonly used interval in literature it may be unnecessary. The behavior of the distribution of the *trans* state is slightly different. Here the middle 50 % of the ω angles distribute between -178.1640 ° and 177.6030 °, respectively. The remaining data is more dense, containing only a few outliers. Hence it can be said, that the interval for the *trans* isomerization state fits quite well onto the purposes of exact feature extraction.

4.2 Optimal Sphere Radius

The results of the analysis of the optimal sphere radius showed, that a sphere radius of 5 Å outperforms other radii (see table 3.4). Within this environment the properties were extracted at their best. The higher the sphere radius was chosen, the less the accuracy was. Therefore it can be assumed, that the 5 Å environment around Xaa-Pro produces the lowest statistical noise. If the sphere radius is set too high, there are amino acids considered, which do not have any influence to the isomerization state of Xaa-Pro. Otherwise, if the radius is chosen to low, there is not enough information present in the environment for feature extraction. A sphere radius of 4 Å can be seen as the allowed minimum. Below this value nearly no amino acids are considered to be in the environment of Xaa-Pro, resulting in undefined features and non-suitable training vectors. For further analysis it may be reasonable to variate the sphere radius between 4 Å and 5 Å in finer steps.

4.3 Structural Feature Importance

The importance of the secondary structure as feature for the Xaa-Pro isomerization was clearly shown. This was also investigated and confirmed in former researches by SONG ET AL. and PAHLKE ET AL. Concerning the real secondary structure and not the predicted secondary structure was used in this work, the information is even more substantial for prediction. The secondary structure of proline achieved the highest average F-score in all datasets (see 3.5). Followed by the secondary structure of the proline preceding amino acid $i-1$ and the secondary structure of the succeeding amino acid $i+1$. It can be assumed that the secondary structure of closely adjacent residues influences the Xaa-Pro isomerization. There may be correlations in respect to positional preferences of amino acids, proposed by REIMER and FISCHER (mentioned in section 1.2), due to amino acid type depending secondary structure formation preferences. Generally there can be seen a slightly higher scoring for the amino acid secondary structure in N-terminal direction relative to proline.

The environment around Xaa-Pro did not reach such a high importance like the secondary structure information. However, a remarkably high F-score achieved the mutability (M_{env}) and the bulkiness (B_{env}) of the environment. Concerning the bulkiness the assumption can be made, that beside the influence of the bulkiness of proline preceding residues (mentioned in section 1.3), the bulkiness of the proline surrounding residues also influences the Xaa-Pro *cis*/*trans* isomerization. The remaining properties yielded no more than the half F-score of the mutability and the bulkiness. It seems that pure physicochemical properties of the Xaa-Pro surrounding environment does not influence the isomerization itself that much. Also the energy approximation of proline and the inside/outside classification performed badly. The model used for the energy calculation was too coarse-grained. Instead of using predetermined statistics, a calculation of a dataset specific inside/outside statistics for energy approximation should be preferred. The inside/outside classification should maybe be replaced by the classical SASA calculation in further researches. This is meaningful because the studies by PAHLKE ET AL. showed, that the *cis* isomerization state occurs more frequent when the proline is exposed to the solvent.

4.3.1 Structural Feature Abstraction Deficiencies

Summarizing it can be said, that the environment of proline influences the Xaa-Pro isomerization. But the methods used for the abstraction of this environment are not yet matured. The use of amino acid property scales is a reasonable approach for the extraction of information. Though the abstraction of the information was not sensitive enough. The use of the density as normalization parameter (see section 2.3) does not perform well and the information gets blurred. There needs to be more information included for the classifier, namely positional information (e.g. Where in the environment

are hydrophobic regions?). But this is non-trivial because the classifier needs discrete numeric values as input. By developing an abstraction method, offering more detailed and additional information of the Xaa-Pro surrounding environment, the prediction accuracy probably can be significantly increased. This can be seen as the main problem, which has to be solved in further attempts.

4.4 Quality of the Parameter Grid-Search Approach

The five-fold cross-validation grid-search results showed the importance of finding the correct SVM kernel parameters. The search interval of growing exponential sequences emphasized as good and practical method and can be used in further researches. The RBF kernel clearly outperformed all other kernels. This can be seen by comparing the mean of the RBF kernel accuracy (red line in figure 3.5) with other kernel functions. Although the polynomial kernel achieved only a slightly less average accuracy over all datasets compared to the RBF kernel (see table 3.6). Unfortunately the grid-search on polynomial kernel functions takes more computational time, depending on the degree d of the kernel. By variation of d probably a higher prediction rate could be achieved. This was done by WANG ET AL. using a polynomial kernel with $d = 8$ based SVM for the prediction of the Xaa-Pro *cis*/*trans* isomerization. A training dataset containing 2026 training vectors, 1013 in *cis* and 1013 in *trans* conformation, was used as basis. They performed a 20-residue local sequence based prediction approach and achieved a prediction accuracy of 70.40 % and 69.70 % for independent testing data and an accuracy of 76.70 % and 76.60 % for cross-validation, respectively [Wang et al., 2004]. Also EXARCHOS ET AL. presented a SVM based prediction method by using for example PSSMs as feature input. They also performed best with a polynomial kernel function, yielding 70.00 % accuracy for the prediction of the peptide bond conformation of any two amino acids [Exarchos et al., 2009]. Therefore this method was not limited to Xaa-Pro fragments.

The linear kernel, reaching only 63.9050 % average accuracy, is not suitable for the prediction of Xaa-Pro isomerization. However, the RBF kernel was the means of choice for a reasonable computation time and a high accuracy of the classifier. This corresponds with other prediction approaches done by SONG ET AL., reaching an accuracy of 71.50 % for a RBF kernel trained classifier based on Position-Specific Iterative Basic Local Alignment Search Tool (PSI-BLAST) profiles and secondary structure information [Song et al., 2006].

A finer grid-search based on smaller search intervals and finer steps of 0.25 and even 0.10 for the RBF kernel, did not result in better prediction accuracy. By searching a finer parameter for a specific model the accuracy of other models decreased, which is not desirable. A main drawback of the grid-search approach for finding the best parameters is the high computational time of this method. But on the other hand it

is easy to parallelize the grid-search because each parameter is independent, which makes it to a common method for finding optimal kernel parameters.

4.5 Assessment of the Prediction Performance

To evaluate the prediction performance of the developed classifiers common statistical methods for the description of classifiers were used. The classifiers trained with the use of the RBF kernel and the parameters $C = 2^{15}$ and $\gamma = 2^{-3}$, emerged to be the best. In ten-fold cross-validation comparison as seen in figure 3.8, it is obvious, that other kernel function did not achieve a comparable accuracy which was discussed in section 4.4. The red line, indicating the overall accuracy of the models trained by using the RBF kernel, exceeds the other models, trained with linear and polynomial kernel, respectively.

In detail the RBF kernel trained models achieved an overall accuracy of 69.8302 %. To be more specific: one model outperformed all others and thus this one was used for the integration in the Xaa-Pro *cis*/*trans* isomerization prediction software μXaa-PIPT. Model 8 emerged as the best classifier with an accuracy of 70.0478 % in ten-fold cross validation. A MCC of 0.4223 was reached, a sensitivity of 0.5433 and a specificity of 0.8576 (see table 3.9). Even if other models achieved higher MCC, sensitivity and specificity, model 8 is a very good compromise for the use as classifier. This is also underlined by an AUC value of 0.7004 (see table 3.8), which was the best of all models. This value indicates a robust and precise classifier for the isomerization state. Nevertheless a higher AUC value would be desirable. Comparable studies by EXARCHOS ET AL. developed classifiers reaching AUC values around 0.90 [Exarchos et al., 2009] where the average AUC value of the RBF kernel trained classifiers in this work only reaches 0.6983.

Very remarkably are the high specificity values, reaching almost 0.86 in average, for all models. These values indicate a high probability for the correct identification of *cis* Xaa-Pro fragments. That means, in case of model 8, that approximately 85 % of the prolines in *cis* conformation are correctly predicted. This indicates a strong correlation between the environment of Xaa-Pro and the *cis* isomerization state. Unfortunately the sensitivity was determined to be around an average value of 0.54, leading to some incorrect classified *trans* prolines. By raising the sensitivity a strong classifier would emerge.

All in all the performance of the trained classifiers could be very well estimated by using the statistical methods describes in section 2.9. The ten-fold cross validation approach against unknown and independent test data, represented the classifiers performance much better than the five-fold cross validation used for the grid-search. This is because the grid-search approach divides every training dataset by five and performs a one-against-the-rest estimation of the prediction accuracy. This leads to the fact, that only a

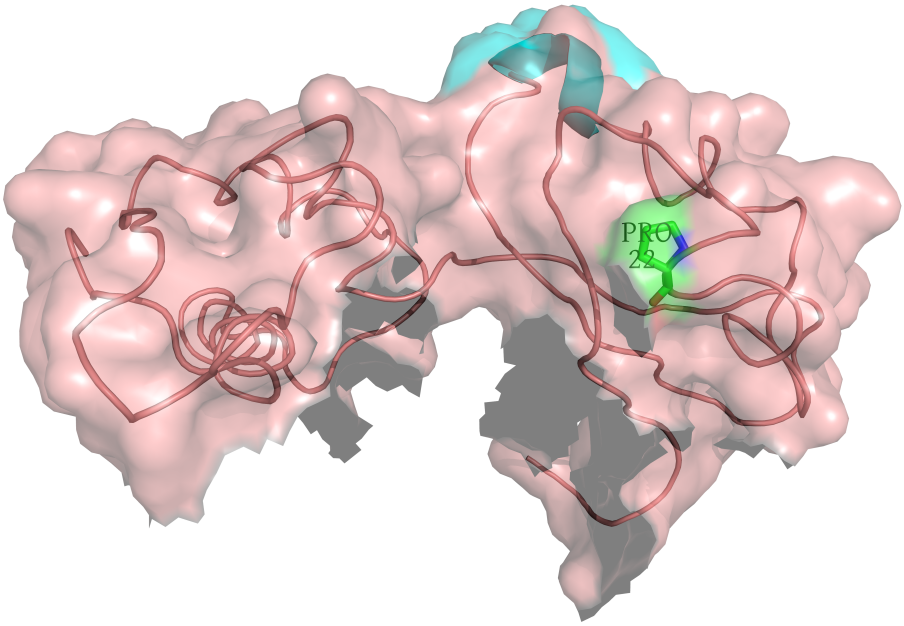

Figure 4.1: Pro22 in *Escherichia coli* K-12 50S ribosomal protein L11 – X-ray diffraction, 3.00 Å resolution, PDB-ID 2R8S

fifth of the originally provided training vectors are used for the training of the classifier. When the complete training datasets were used, much more powerful classifiers were generated. This could be even carried forward by using a larger dataset for the training. This could be achieved by changing the criteria for the creation of the initial dataset.

4.5.1 Case Study

To present a case study the application of μXaa-PIPT is demonstrated at the N-terminal domain (NTD) of the *Escherichia coli* K-12 50S ribosomal protein L11. Recent studies of WANG ET AL. in 2012 figured out, that the *cis* conformation of conserved Pro22 in L11 (see figure 4.1) is essential for the interaction between the L11 NTD and the L12 CTD. The PPIase activity of the elongation factor G (EF-G) drives the *cis*/*trans* isomerization of Pro22 and thus the protein synthesis and cell survival in *Escherichia coli* [Wang et al., 2012].

There are several structures of different quality resolved of the 50S ribosomal protein L11, which is coded by the gene *rplK*. Using the UniProt Knowledgebase (UniProtKB) database the corresponding PDB structures were identified. The analysis of Pro22 with μXaa-PIPT in five of these structures of different resolution resulted in the prediction of the *cis* state in all cases, whereas only one structure (2GYA_G) assigned the *cis* state to Pro22 (see table 4.1). Thus the environment around proline and the structural information clearly indicates the *cis* isomerization state. The leads to the conclusion, that μXaa-PIPT performs well in the identification of *cis* prolines with possible biological

Table 4.1: μXaa-PIPT prediction of the isomerization state of Pro22 in *Escherichia coli* K-12 50S ribosomal protein L11, structures of different resolutions

PDB-ID, chain ID	experimental type	resolution in Å	Pro22 structure	ω angle in degree	Pro22 predicted
3R8S_I	X-ray	3.00	trans	-177.5647	cis
IVS6_I	X-ray	3.46	trans	-179.9919	cis
2Z4L_I	X-ray	4.00	trans	179.9857	cis
2J28_I	electron microscopy	8.00	trans	179.8848	cis
3J0D_G	electron microscopy	11.10	trans	179.8137	cis
2GYA_G	electron microscopy	15.00	cis	0.0788	cis

functions. This method can be applied on other structures with prolines, that are considered to be of functional interest, to gain more evidences and to evaluate existing protein structures.

4.6 Investigation of the Intermediate State

By applying the best classifier (model 8) to Xaa-Pro fragments with an intermediate isomerization state, worthwhile results were produced. As seen in table 3.10, 160 of the prolines (80.4020 %) were predicted to be in *cis* isomerization state. Whereas only 39 prolines (19.5980 %) were classified to be in the *trans* conformation and 6 prolines (2.2968 %) were unpredictable because of incomplete features. The result was not expected because the classifier does not have such a significant rate of FN predictions to prefer the *cis* class.

This result is extraordinary as it indicates that the surrounding environment of the prolines in *cis*/*trans* intermediate state clearly indicates that they are in *cis* conformation. These specific prolines may interacting as fundamental molecular switch in the protein. LU ET AL. proposed that the conversion from *trans* to *cis* state happens more often than vice versa. So there can be assumed that at least some of the prolines classified as *cis* do have a biological background. Further investigations should be done, to identify possible functional Xaa-Pro fragments and analyze the specific prolines in detail.

4.7 Conclusion

A new method for the prediction of the Xaa-Pro isomerization state using structural features was developed. Based on the SVM learning technique, there were accuracies achieved, comparable to those of EXARCHOS ET AL., SONG ET AL. and PAHLKE ET AL. The studies in this work confirm one more time the power of SVMs for the prediction of the prolyl peptide bond conformation.

The software Xaa-PIPT for the extraction and abstraction of structural features around was developed. Beside this a lightweight prediction tool – μXaa-PIPT – was developed for the use by end users. This tool can be used by researchers to predict and evaluate

the isomerization state of Xaa-Pro in structures of low resolution or theoretical models, based on the proline surrounding environment and real structural information. Further applications may include the analysis of specific prolines in respect to their biological functionality. It can be confirmed, that the properties of the amino acids around proline do have influences on the isomerization process and helped to built a classifier with an prediction accuracy of 70.0478 %. But it was also shown that the methods of structural feature extraction in this work are not the proof of concept. The information is too coarse and blurred by the abstraction to numeric values.

4.8 Perspective

In further researches it should be investigated how the structural information could be extracted without being too much abstracted and blurred. Based on a better method than the one presented in this work, a powerful method of Xaa-Pro *cis*/*trans* isomerization classification can be evolved. It will be also of great interest to investigate the rare occurring *cis*/*trans* intermediate state, which is still mainly unknown in function. Further implementations for feature extraction methods or the implementation of other features is straightforward, as the developed software is easy expandable. Maybe multi-class SVM support would be a nice benefit for the prediction of the *cis*/*trans* intermediate state. However, the imbalance of the classes will remain a problem. Maybe it can be resolved using a weighting parameter w for the classes when training the SVM, but it will still be hard to gain enough training vectors of the very rare occurring *cis*/*trans* intermediate state.

Appendix A: Software CD Content and Instructions

- `document/`
  - `fkaiserBachelorThesis.pdf`
- `software/`
  - `XaaPIPT.jar`
  - `MicroXaaPIPT.jar`
  - sources, licenses, etc.
- `dataset/`
  - `all_list.txt`
- `training_data/`
  - `svm_raw.pipt`
  - `svm_scaled.pipt`
  - `svm_balanced_1.pipt`
  - `svm_balanced_2.pipt`
  - `svm_balanced_3.pipt`
  - `svm_balanced_4.pipt`
  - `svm_balanced_5.pipt`
  - `svm_balanced_6.pipt`
  - `svm_balanced_7.pipt`
  - `svm_balanced_8.pipt`
  - `svm_balanced_9.pipt`
  - `svm_balanced_10.pipt`
- `testing_data/`
  - `svm_balanced_1.pipt.test`
  - `svm_balanced_2.pipt.test`
  - `svm_balanced_3.pipt.test`
  - `svm_balanced_4.pipt.test`
  - `svm_balanced_5.pipt.test`
  - `svm_balanced_6.pipt.test`

  - `svm_balanced_7.pipt.test`
  - `svm_balanced_8.pipt.test`
  - `svm_balanced_9.pipt.test`
  - `svm_balanced_10.pipt.test`

- `svm_models/` (RBF kernel trained models with $C = 2^{15}$ and $\gamma = 2^{-3}$)

  - `svm_balanced_1.pipt.model`
  - `svm_balanced_2.pipt.model`
  - `svm_balanced_3.pipt.model`
  - `svm_balanced_4.pipt.model`
  - `svm_balanced_5.pipt.model`
  - `svm_balanced_6.pipt.model`
  - `svm_balanced_7.pipt.model`
  - `svm_balanced_8.pipt.model`
  - `svm_balanced_9.pipt.model`
  - `svm_balanced_10.pipt.model`

To launch Xaa-PIPT or μXaa-PIPT:

- On Windows systems: double click the `XaaPIPT.jar` or the `MicroXaaPIPT.jar` file
- On Unix-like systems: execute the shell command `java -jar XaaPIPT.jar` or `java -jar MicroXaaPIPT.jar`

The Java Runtime Environment (JRE) needs to be installed to run the software.

Appendix B: Software Screenshots

The following figures show the GUI of the Java software Xaa-PIPT, that was developed for this work.

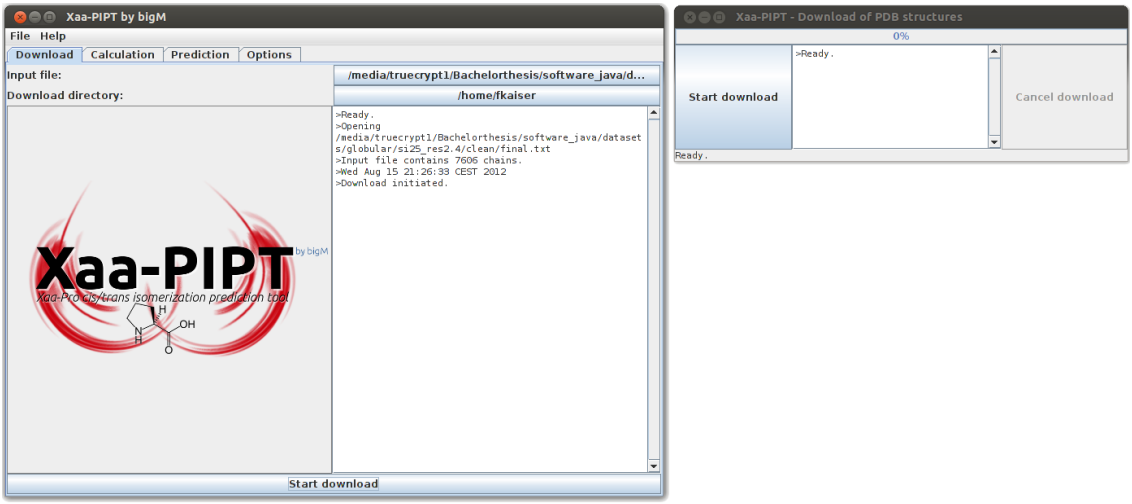

Figure B.1: Xaa-PIPT download interface

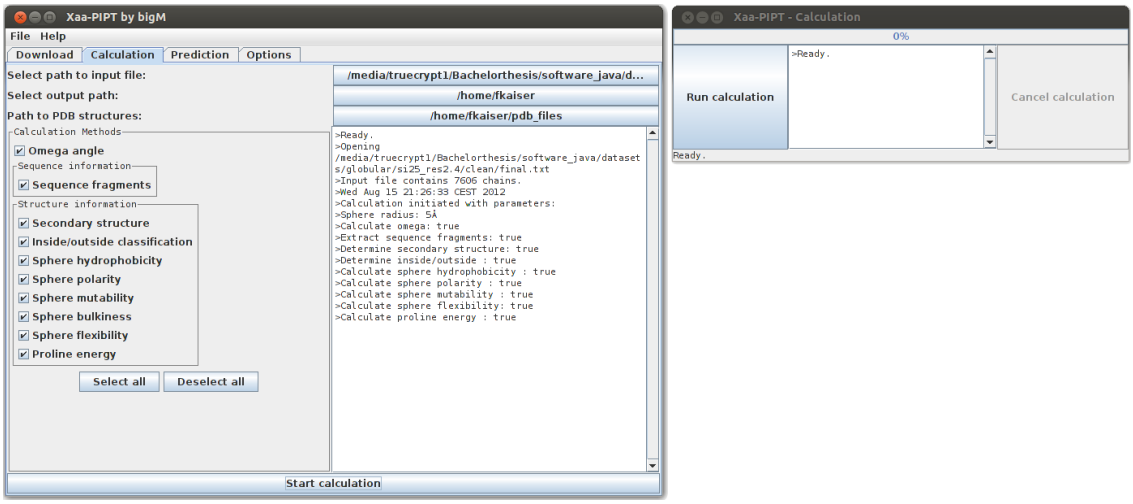

Figure B.2: Xaa-PIPT calculation interface

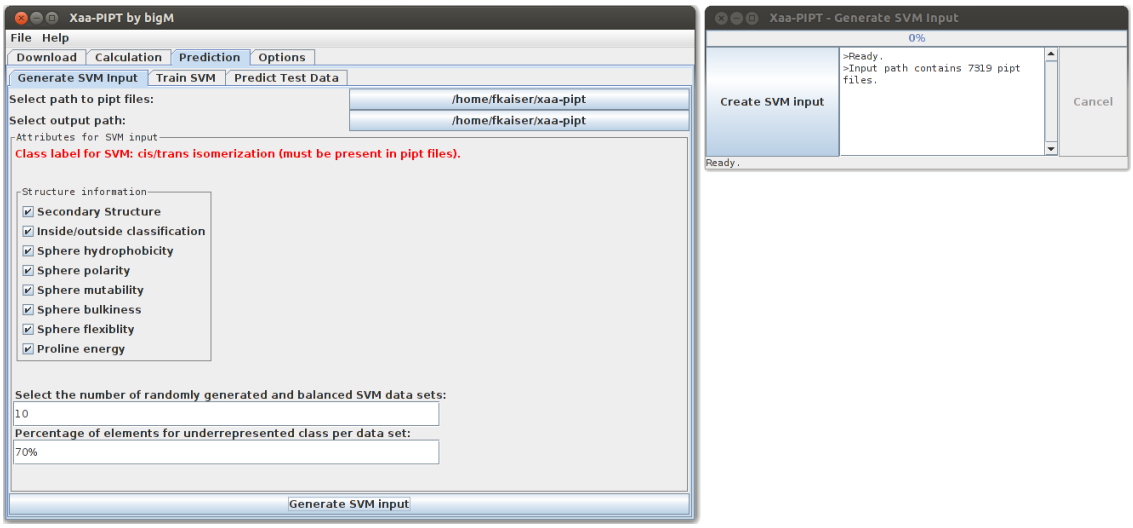

Figure B.3: Xaa-PIPT SVM input generation interface

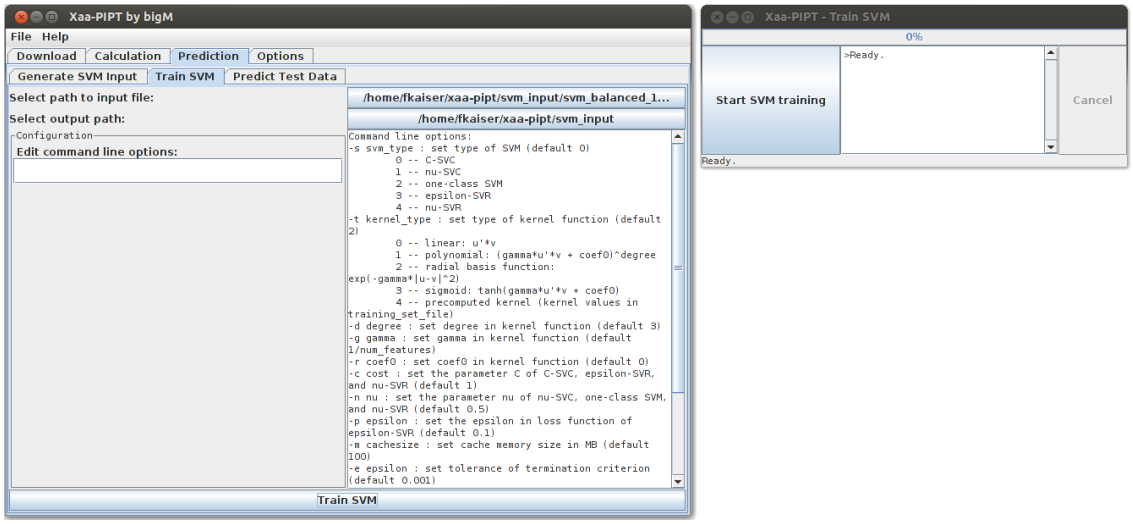

Figure B.4: Xaa-PIPT SVM training interface

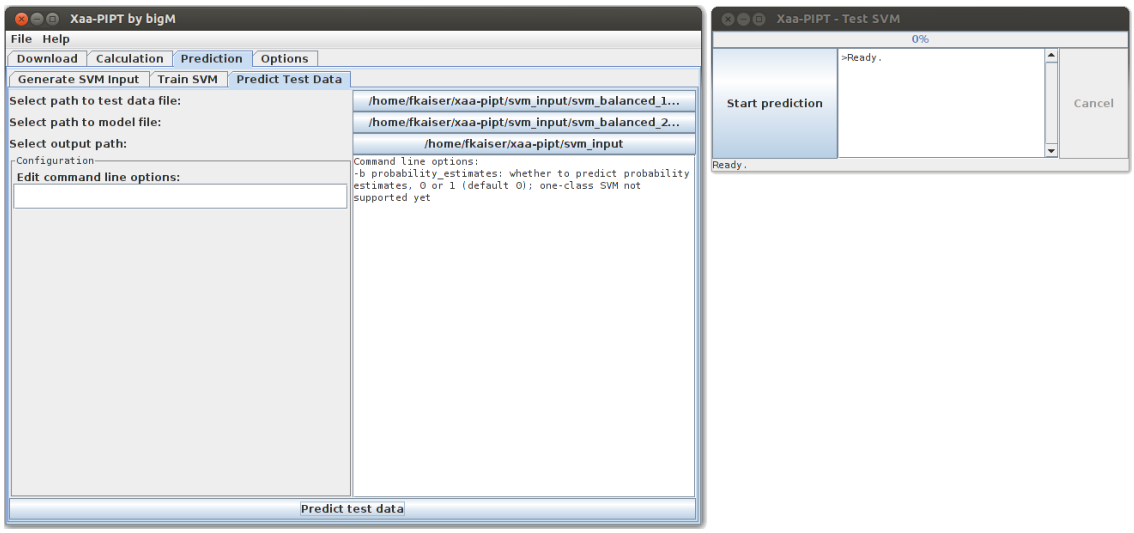

Figure B.5: Xaa-PIPT SVM prediction interface

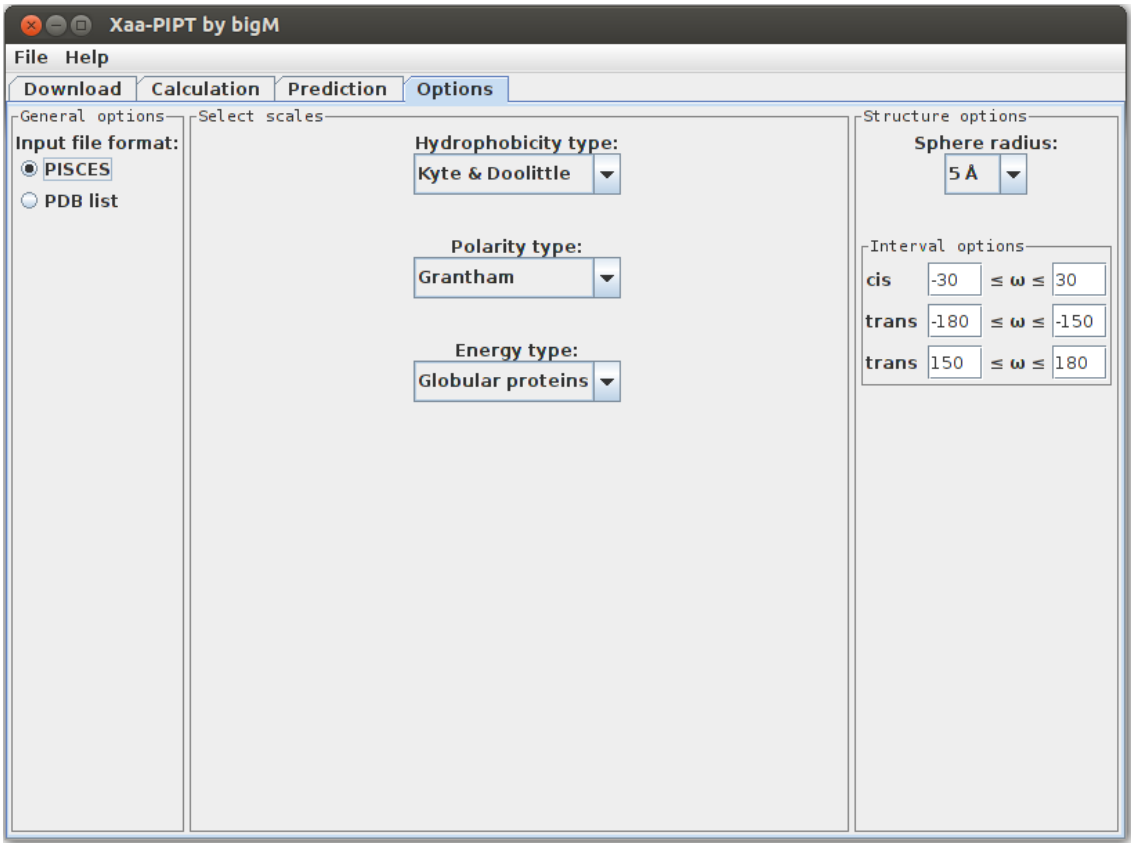

Figure B.6: Xaa-PIPT options interface

Bibliography

[Abe, 2005a] Abe, S. (2005a). Multiclass support vector machines. In *Support Vector Machines for Pattern Classification*, Advances in Pattern Recognition, pages 83–128. Springer London.

[Abe, 2005b] Abe, S. (2005b). Two-class support vector machines. In *Support Vector Machines for Pattern Classification*, Advances in Pattern Recognition, pages 15–82. Springer London.

[Bhaskaran and Ponnuswamy, 1988] Bhaskaran, R. and Ponnuswamy, P. (1988). Positional flexibilities of amino acid residues in globular proteins. *International Journal of Peptide and Protein Research*, 32(4):241–255.

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[Chen and Lin, 2006] Chen, Y.-W. and Lin, C.-J. (2006). Combining svms with various feature selection strategies. In Guyon, I., Nikravesh, M., Gunn, S., and Zadeh, L., editors, *Feature Extraction*, volume 207 of *Studies in Fuzziness and Soft Computing*, pages 315–324. Springer Berlin / Heidelberg.

[Chih-Wei et al., 2010] Chih-Wei, H., Chih-Chung, C., and Chih-Jen, L. (2010). A practical guide to support vector machine classification. *Department of Computer Science, National University Taiwan, Taipei 106, Taiwan*, page 16.

[Exarchos et al., 2009] Exarchos, K. P., Papaloukas, C., Exarchos, T. P., Troganis, A. N., and Fotiadis, D. I. (2009). Prediction of cis/trans isomerization using feature selection and support vector machines. *J Biomed Inform*, 42(1):140–149. [DOI:10.1016/j.jbi.2008.05.006] [PubMed:18586558].

[Grantham, 1974] Grantham, R. (1974). Amino acid difference formula to help explain protein evolution. *Science*, 185(4154):862–864. [PubMed:4843792].

[Grathwohl and Wüthrich, 1981] Grathwohl, C. and Wüthrich, K. (1981). Nmr studies of the rates of proline cis-trans isomerization in oligopeptides. *Biopolymers*, 20:2623–33.

[Heinke and Labudde, 2012] Heinke, F. and Labudde, D. (2012). Membrane protein stability analyses by means of protein energy profiles in case of nephrogenic diabetes in-

sipidus. *Comput Math Methods Med*, 2012:790281. [PubMed Central:PMC3312259] [DOI:10.1155/2012/790281] [PubMed:22474537].

[Holland et al., 2008] Holland, R. C., Down, T. A., Pocock, M., Prlic, A., Huen, D., James, K., Foisy, S., Drager, A., Yates, A., Heuer, M., and Schreiber, M. J. (2008). BioJava: an open-source framework for bioinformatics. *Bioinformatics*, 24(18):2096–2097. [PubMed Central:PMC2530884] [DOI:10.1093/bioinformatics/btn397] [PubMed:18689808].

[Kecman, 2005] Kecman, V. (2005). Support vector machines – an introduction. In Wang, L., editor, *Support Vector Machines: Theory and Applications*, volume 177 of *Studies in Fuzziness and Soft Computing*, pages 1–49. Springer Berlin / Heidelberg.

[Kyte and Doolittle, 1982] Kyte, J. and Doolittle, R. F. (1982). A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.*, 157(1):105–132. [PubMed:7108955].

[Labudde et al., 2012] Labudde, D., Heinke, F., Schildbach, S., and Stockmann, D. (2012). Energy profile suite (ePros). online at http://bioservices.hs-mittweida.de/Epros/Index.

[Lee and Richards, 1971] Lee, B. and Richards, F. M. (1971). The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.*, 55(3):379–400. [PubMed:5551392].

[Lu et al., 2007] Lu, K. P., Finn, G., Lee, T. H., and Nicholson, L. K. (2007). Prolyl cis-trans isomerization as a molecular timer. *Nat. Chem. Biol.*, 3(10):619–629. [DOI:10.1038/nchembio.2007.35] [PubMed:17876319].

[Lummis et al., 2005] Lummis, S. C., Beene, D. L., Lee, L. W., Lester, H. A., Broadhurst, R. W., and Dougherty, D. A. (2005). Cis-trans isomerization at a proline opens the pore of a neurotransmitter-gated ion channel. *Nature*, 438(7065):248–252. [DOI:10.1038/nature04130] [PubMed:16281040].

[Margaret O. Dayhoff, 1978] Margaret O. Dayhoff, R. M. Schwartz, B. C. O. (1978). *Atlas of protein sequence and structure*, volume 5. National Biomedical Research.

[Pahlke et al., 2005] Pahlke, D., Freund, C., Leitner, D., and Labudde, D. (2005). Statistically significant dependence of the Xaa-Pro peptide bond conformation on secondary structure and amino acid sequence. *BMC Struct. Biol.*, 5:8. [PubMed Central:PMC1087856] [DOI:10.1186/1472-6807-5-8] [PubMed:15804350].

[Reimer and Fischer, 2002] Reimer, U. and Fischer, G. (2002). Local structural changes

caused by peptidyl-prolyl cis/trans isomerization in the native state of proteins. *Biophys. Chem.*, 96(2-3):203–212. [PubMed:12034441].

[Shrake and Rupley, 1973] Shrake, A. and Rupley, J. A. (1973). Environment and exposure to solvent of protein atoms. Lysozyme and insulin. *J. Mol. Biol.*, 79(2):351–371. [PubMed:4760134].

[Sing et al., 2005] Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (2005). ROCR: visualizing classifier performance in R. *Bioinformatics*, 21(20):3940–3941. [DOI:10.1093/bioinformatics/bti623] [PubMed:16096348].

[Song et al., 2006] Song, J., Burrage, K., Yuan, Z., and Huber, T. (2006). Prediction of cis/trans isomerization in proteins using PSI-BLAST profiles and secondary structure information. *BMC Bioinformatics*, 7:124. [PubMed Central:PMC1450308] [DOI:10.1186/1471-2105-7-124] [PubMed:16526956].

[Wang and Dunbrack, 2003] Wang, G. and Dunbrack, R. L. (2003). PISCES: a protein sequence culling server. *Bioinformatics*, 19(12):1589–1591. [PubMed:12912846].

[Wang et al., 2012] Wang, L., Yang, F., Zhang, D., Chen, Z., Xu, R. M., Nierhaus, K. H., Gong, W., and Qin, Y. (2012). A conserved proline switch on the ribosome facilitates the recruitment and binding of trGTPases. *Nat. Struct. Mol. Biol.*, 19(4):403–410. [DOI:10.1038/nsmb.2254] [PubMed:22407015].

[Wang et al., 2004] Wang, M. L., Li, W. J., Wang, M. L., and Xu, W. B. (2004). Support vector machines for prediction of peptidyl prolyl cis/trans isomerization. *J. Pept. Res.*, 63(1):23–28. [PubMed:14984570].

[Wedemeyer et al., 2002] Wedemeyer, W. J., Welker, E., and Scheraga, H. A. (2002). Proline cis-trans isomerization and protein folding. *Biochemistry*, 41(50):14637–14644. [PubMed:12475212].

[Weiss et al., 1998] Weiss, M. S., Jabs, A., and Hilgenfeld, R. (1998). Peptide bonds revisited. *Nat. Struct. Biol.*, 5(8):676. [DOI:10.1038/1368] [PubMed:9699627].

[Zimmerman et al., 1968] Zimmerman, J. M., Eliezer, N., and Simha, R. (1968). The characterization of amino acid sequences in proteins by statistical methods. *J. Theor. Biol.*, 21(2):170–201. [PubMed:5700434].

Glossary

cis/trans A state of isomerism between the *cis* and the *trans* state.

cis A form of conformational isomerism. Two considered molecule residues standing on the same side in relation to a reference plane.

trans A form of conformational isomerism. Two considered molecule residues towards each other in relation to a reference plane.

amino acid property scales Scales allocating every amino acid numeric values according to specific properties. A summary of different scales is located at http://web.expasy.org/protscale/.

BioJava A powerful and essential open source framework in bioinformatics. It offers analysis tools, statistical routines and other useful tools for the processing of biological data. The project website can be found on http://biojava.org/wiki/Main_Page.

boxplot A graphical representation of the distribution of data, showing the quartiles, the median, the maximum and the minimum together with possible outliers.

CUDA An computing architecture developed by Nvidia for the acceleration of computationally intensive calculations on graphic devices.

Eclipse A very common open source SDK mainly for Java related applications and C/C++ developers. The project homepage can be found on http://www.eclipse.org/.

Energy Profile Suite A database and toolbox for calculation, analysis, comparison and prediction of protein energy profiles, available at http://bioservices.hs-mittweida.de/.

F-score A simple but effective method for the estimation of feature importance and selection.

FASTA format A text based format for the representation of nucleotide or protein sequences with single letter codes. The first line starts with ">" and contains a annotation of the sequence which follows after a line break.

feature Input for a classifier are called features. They are determined so, that they represent each class as best as possible and the classes are well separated in the input space.

libSVM An integrated software library for support vector classification with multi-class classification support. The library is available for a variety of programming languages and thus multi-platform support is given. The package can be found on http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Matthews correlation coefficient A measurement for the performance of a binary classifier resulting from a confusion matrix. Also known as Φ correlation coefficient.

PISCES A web server for culling PDB files regarding specified parameters (sequence identity, resolution, etc.). It can be found on http://dunbrack.fccc.edu/PISCES.php/.

Protein Data Bank of Transmembrane Proteins A data bank redundant to the PDB, containing only structures of transmembrane proteins (http://pdbtm.enzim.hu/).

R An open-source programming language for statistical calculations and graphics. Available at http://www.r-project.org/.

R-factor A statistical measurement of the agreement between a model of a structure and its X-ray diffraction data ranging from $-1 \leq R \leq 1$.

receiver operating characteristics curve A graphical representation of the relationship between TP and FP prediction rate. The area under the ROC curve (AUC) is an important value measuring the prediction performance of a classifier.

solvent accessible surface area The for a specific solvent defined accessible area of a biomolecule. The solvent accessible surface area is usually described in Å^2.

support vector machine A machine learning technique, which can be used for classification and regression of unknown data.

UniProt Knowledgebase A database containing high-quality functional information of proteins. Accessible over http://www.uniprot.org/.

Xaa-Pro A dipeptide consisting of proline (i) and the preceding amino acid ($i-1$), where Xaa can be any amino acid.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, 17.08.2012